10/715,878 Paper 20050913 Reference N

WO 01/19005 PCT/US00/24405

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 March 2001 (15.03.2001)

PCT

(10) International Publication Number
WO 01/19005 A1

(51) International Patent Classification[7]: H04J 3/06, H04N 7/173, H04L 12/28

(21) International Application Number: PCT/US00/24405

(22) International Filing Date: 5 September 2000 (05.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/152,254 3 September 1999 (03.09.1999) US

(71) Applicant (for all designated States except US): BROADCOM CORPORATION [US/US]; 16215 Alton Parkway, Irvine, CA 92618-3616 (US).

(72) Inventors; and
(75) Inventors/Applicants (for US only): RABENKO, Theodore, F. [US/US]; 16215 Alton Parkway, Irvine, CA 92618-3616 (US). DENNEY, Lisa, V. [US/US]; 16215 Alton Parkway, Irvine, CA 92618-3616 (US).

(74) Agent: GELFOUND, Craig, A.; Christie, Parker & Hale, LLP, P.O. Box 7068, Pasadena, CA 91109-7068 (US).

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*
— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 01/19005 A1

(54) Title: SYSTEM AND METHOD FOR THE SYNCHRONIZATION AND DISTRIBUTION OF TELEPHONY TIMING INFORMATION IN A CABLE MODEM NETWORK

(57) Abstract: A method for synchronizing clocks in a packet transport network. The method comprises, receiving an external network clock at a central packet network node and transmitting timing information to a plurality of packet network devices, the timing information based upon the external network clock. The method further comprises, transmitting and receiving data that is synchronized to the timing information to a plurality of connected packet network devices. And finally, delivery of packets to an external interface via a packet network that contains data synchronized to the external network clock.

SYSTEM AND METHOD FOR THE SYNCHRONIZATION AND DISTRIBUTION OF TELEPHONY TIMING INFORMATION IN A CABLE MODUM NETWORK

BACKGROUND OF THE INVENTION

Circuits for the distribution and synchronization of timing information play a key role in a number of applications which require a synchronus transfer of data, such as networks for transferring telephone calls over various networks, including the internet, and the like.

Current methods of signal synchronization between sub-networks do not provide complete synchronization. Incomplete synchronization results in data losses called slips. Compensating networks, including buffer circuitry, are typically used to compensate for slips caused by a lack of clock synchronization.

Those having skill in the art will understand the desirability of having a completely synchronous timing of sample collection and reconstruction that eliminates slips and the need for compensating circuitry.. This type of network would provide complete synchronization of clocks between sub-networks by providing a series of clocks slaved to a master clock.

SUMMARY OF THE INVENTION

There is therefore provided in a present embodiment of the invention a method for synchronizing clocks in a packet transport network. The method comprises,receiving an external network clock at a central packet network node and transmitting timing information to a plurality of packet network devices, the timing information based upon the external network clock.

The method further comprises, transmitting and receiving data that is synchronized to the timing information to a plurality of connected packet network devices. And finally, delivery of packets to an external interface via a packet network that contains data synchronized to the external network clock.

Many of the attendant features of this invention will be more readily appreciated as the same becomes better understood by reference to the following detailed description considered in connection with the accompanying drawings.

DESCRIPTION OF THE DRAWINGS

These and other features and advantages of the present invention will be better understood from the following detailed description read in light of the accompanying drawings, wherein:

FIG. 1 is an illustration of a network system having synchronous clocking of digital telephony data between a Public Switched telephone network (PSTN) and an internet network via a gate way;

FIG. 2 is a block diagram of an internet telephone transmission system 1002 utilizing a cable television (CATV) network to couple one or more telephones that are in communications with each other; and

FIG. 3 is an illustration of an embodiment of a system for the synchronization and distribution of a fully synchronized clock signal.

Like reference numerals are used to designate like parts in the accompanying drawings.

DETAILED DESCRIPTION OF THE INVENTION

FIG. 1 is an illustration of a network system 0001 having synchronous clocking of voice telephony data between a telephone 1001 coupled to a conventional Public Switched Telephone Network (PSTN) 1000 and a telephone coupled a Digital Data Transport Network 2000. The telephone 1001 coupled to the PSTN uses conventional and ubiquitous interface methods typically used in virtually every home and business in North America today. The telephone 2001 coupled to the Digital Data Transport Network 2000 is capable of being coupled in any of a variety of methods in use today to include, but not limited to Voice over Internet Protocol (VoIP) or Voice over Digital Subscriber Loop (VoDSL).

For the purpose of this example, two telephones 1001 and 2001 are assumed to be identical. However, equivalent devices are available and interchangable including ISDN phone or evolving Ethernet or VoIP phone instruments that provide equivilent functions. Those skilled in the art will recognize that the description of the interfaces and functions that follow are one of many equivilent configurations that are used to practice the described embodiment.

The interface between the telephone 1001 and the PSTN 1000 is a conventional loop start interface as described in the Telcordia document TR-NWT-000057. The interface between the PSTN 1000 and the Station Reference is a conventional Building Integrated Timing System 1020 (BITS) as described in Telcordia TR-NWT-001244. The interface between the PSTN 1000 and the Gateway 1050 is a conventionalGR-303 interface 1030.

The interface 1065 between the Station reference 1040 and the Gateway 1050 is a BITS interface. The interface between the Station Reference 1040 and the Data Transport Network is the well known Data-Over-Cable Service Specification (DOCSIS) as specified by CableLabs in SP-RFI-I04-0980724. The interface between the Gateway 1050 and the Data Transport Network 2000 is the well known IEEE 802.3 interface a.k.a. Ethernet. The interface between the Cable Modem 2300 and the Data Transport Network is the well known DOCSIS interface. The interface between the Cable Modem 2300 and the telephone 2001 is the loop start interface as described in TR-NWT-000057.

All of the interfaces used in the practice of this invention are standards based and well known to those skilled in the art. Traditional implementations of Gateway devices between the PSTN and Data Transport networks ignore the timing information provided by the PSTN. The consequence of this design practice is that it tends to introduce large delay and data loss to the voice signal at the gateway thereby compromising the quality of the voice signal.

The present embodiment of the invention provides a system and a method of delivering the PSTN timing information using data transport methods so that the sampling and playout of voice information at the Gateway 1050 and the Cable Modem 2300-2001 is performed synchronously. The synchronous operation of the embodiments of the invention minimizes data loss and the total delay experienced by the voice data as it is transported through the Data Transport Network.

FIG. 2 is a block diagram of an internet telephone transmission system 1002 utilizing a cable television (CATV) network 1026 to couple one or more telephones 2002, 2008, 2010

that are in communications with each other. The networks described in FIG. 2 are more fully described in Appendix 1 amd Appendix 2.

In the embodiment shown a telephone 2002 is coupled to a PSTN 1004 in a conventional manner known to those skilled in the art. The PSTN 1004 is coupled to an ISP Gateway 1012. Typically the PSTN to Gateway connection utilizes digital signal transmission as known to those skilled in the art. The ISP gateway 1012 is coupled to an internet 1006 utilizing conventional signal transmission protocols known to those skilled in the art.

The internet1006 is coupled to a CATV network 1026. The CATV network comprises a cable modem termination system (CMTS) 2004, a hybrid fiber coax (HFC) network 1010, and a cable modem 2006. The CMTS 2004 is coupled to the internet 1006 in a conventional manner known to those skilled in the art. The CMTS 2004 is coupled to the HFC 1010 in a conventional manner known to those skilled in the art. The HFC 1010 is coupled to the cable modem 2006 in a conventional manner known to those skilled in the art.

The cable modem 2006 is used as an access point to couple other networks, such as an HPNA network 1014, and other devices such as a PC 2012, and a telephone 2010 to the internet 1006. A PC 2012 is coupled to the cable modem 2006 in a conventional manner known to those skilled in the art. A television, or video system 2014 is coupled to the cable modem 2006 in a conventional manner known to those skilled in the art. A telephone 2010 is coupled to the cable modem 2006 in a conventional manner known to those skilled in the art.

The Cable modem 2006 is also coupled to an external network such as an HPNA network 1014 in a conventional manner known to those skilled in the art. The HPNA network shown comprises a HPNA phone adapter 2016. The cable modem 2006 is coupled to the HPNA Phone adapter 2016 in a conventional manner known to those skilled in the art. The HPNA phone adapter is coupled to a conventionally constructed telephone 2008 in a conventional manner known to those skilled in the art. The transmission system, utilizing the cable television network 1026, typically

enables a home computer user to network their computer 2012 to the internet 1006 through a cable TV transmission network 1026through a cable modem 2006. And also a user may make telephone calls through a cable modem 2006 as well as receive television broadcasts on a television set 2014.

The transmission of data over the cable television network 1026 is governed by the Data-Over-Cable Service Interface Specification (DOCSIS). In particular the DOCIS specification SP-RFI-I04-980724 is relevant to the implementation of the embodiments of the invention and is incorporated in its entirety by reference into the text of this application.

Transmission of digital telephony data between telephones 2008 in a home network, or equivalently a locally based network 1014, and over the cable television network 1026 to users not directly coupled to the home network 2002 is governed by a HPNA specification 2.0, incorporated herein in its entirety by reference. Thus, because of increasing use of network systems for telephone traffic, utilization of fully synchronous clocking is becoming more important as the demand to transmit voice over a data network increases.

FIG. 3 is an illustration of an embodiment of a system for the distribution of PSTN timing information signals using data transmission techniques. The collection of coupled networks 2060, 2070, 2080, 2090 forms an overall data transport network 2000 in which timing and voice data signals are transported between the PSTN 1000 and Voice Sampling circuits 2310 and 2410 coupled by the Data Transport Network 2000.

The CMTS 2010 is configured to allow the DOCSIS network clock 2012 to be synchronized to the station reference 1040 by using a well known Stratum 3 reference clock 2011. The performance of the Stratum 3 reference clock is defined by Telcordia TR-NWT-001244. Those skilled in the art will recognize that the synchronization interface 2014 between the Stratum 3 reference and the CMTS master oscillator 2012 is a conventionally constructed Phase Lock Loop (PLL) circuit, as known to those skilled in the art. In the embodiment shown, a CMTS 2010

comprises a Stratum 3 reference clock 2011 coupled 2014 to a CMTS master oscillator 2012 using a PLL circuit. The CMTS master oscillator 2012 is coupled to a DOCSIS head end controller 2013. The DOCSIS head end controller is conventionally constructed as is known to those skilled in the art. An example of this device is the commercially available BCM3210 from Broadcom Corporation. The DOCSIS head end controller 2013 couples an HFC 2060 via an upstream and downstream path 2050, to a QoS managed Ethernet 2090 . The CMTS performs a media conversion operation between the DOCSIS RF network and the Ethernet. This operation is described by SP-RFI-I04-980724. The station reference 1040 and the Stratum 3 reference clock 2011 are conventionally constructed as is known to those skilled in the art.

The Hybrid Fiber Coax (HFC) network 2060 is conventionally constructed as is known to those skilled in the art. The HFC 2060network provides physical transmission between the CMTS 2010 and a cable modem 2300. The DOCSIS data transmission method 2050 & 2200 provides a way to deliver Internet Protocol formatted packets imbedded in MPEG frames. A description of this method is described in SP-RFI-I04-980724. DOCSIS also identifies a method to transmit the CMTS timing master information 2012, using a DOCSIS specific method, to the Cable Modem 2300. The transmission of the clock information 2040 & 2100 permits the Cable Modem to generate a Timing Recovered Clock (TRC) 2312 that is frequency locked to the CMTS Master clock 2012. This embodiment causes the DOCSIS TRC clock 2312 to be frequency locked to the Station Reference 1040.

The cable modem 2300 comprises a DOCSIS CPE controller coupled to a voice sampling circuit 2310 that is in turn coupled to a conventionally constructed external telephone set 2001. The cable modem is conventionally constructed as is known to those skilled in the art. The Cable Modem TRC 2312 is coupled to the Voice Sampling circuit by conventional methods including clock dividers, as needed to match the rate of the TRC to that required by the Voice Sampling Circuit 2310. An example of the DOCSIS CPE Controller is the BCM3350 from Broadcom Corporation. An example

of the Voice Sampling Circuit is the Am79Q031 from Advanced Micro Devices.

The HPNA controller is coupled to the TRC clock by the DOCSIS CPE controller. The HPNA controller provides a method to transmit the TRC timing information using HPNA protocol signals. This circuit is provided as an example to demonstrate that this timing transmission method may be used to further extend the timing network beyond the Cable Modem.

The HPNA controller 2311 of the cable modem serves to couple the HPNA network 2070 to the Ethernet 2090 using the data transport methods provided by the DOCSIS network. The HPNA controller and HPNA network are conventionally constructed as is known to those skilled in the art. The HPNA controller 2311 of the cable modem is coupled 2070 to an HPNA controller 2411 included in an HPNA phone adapter 2400. The HPNA controller 2311 provides a method to transmit the TRC clock 2312 to the HPNA Phone adapter clock 2412 over a messaging interface 2070. The HPNA controller 2411 is coupled to a local clock 2412, and a voice sampling circuit 2410. The voice sampling circuit 2410 is in turn coupled to a conventionally constructed external telephone set 2002.

The PSTN 1000 is conventionally constructed as is known to those skilled in the art. Gateway 1050 is also coupled to the Ethernet 2090, and the PSTN 1000. The PSTN is in turn coupled to a plurality of conventionally constructed telephone sets represented by a single phone 1001.

A cable modem termination system (CMTS) reference 2011 is synchronized to the network station reference 1020. The Station reference 1020 is used to synchronize the internal Stratum 3 reference clock 2011, contained in both the CMTS 2010 and the PSTN Gateway 1050. The Stratum 3 reference clock in the PSTN Gateway 1050 is conventionally constructed as is known to those skilled in the art. The DOCSIS CMTS reference 2012 is slaved to the Stratum 3 reference clock by a Phase Locked Loop (PLL) circuit 2014 .

The DOCSIS CMTS reference 2012 that is synchronized to the PSTN station reference 1040 is transported over the HFC network 2060 to the DOCSIS CPE controller 2313 located in a remote cable modem 2300 using a DOCSIS SYNC method well known to those skilled in the art. The DOCSIS SYNC method causes the DOCSIS CPE controller's clock 2312, to be frequency locked to the CMTS reference clock 2012 which is in turn phase locked to the station reference 1040, which is phase locked to the PSTN clock as provided by the PSTN Clock distribution network. The end result of this connection method is that the DOCSIS CPE Controller's clock 2312, is frequency locked to the PSTN timing distribution network as reflected in the station reference 1040.

At the cable modem 2300, the DOCSIS CPE Controller's Clock 2312, is used to provide timing to voice circuit 2310 that is a part of the cable modem 2300 (or equivalently, are locally coupled to the DOCSIS CPE controller 2313). The DOCSIS CPE Controller's Clock 2312, is also used to provide timing for remotely coupled voice circuits 2411 coupled to an in home network 2080 . The present invention connects these remote voice circuits via a conventional Home PNA network. Those skilled in the art will recognize that this connection may be equivalently accomplished by other network means including conventional Ethernet and Token Ring networks. The network connection is not limited to wired methods, as wireless networks provide an equivilent connection under the operation of various standards including the BlueTooth, IEEE 802.11a/b or HomeRF.

The HPNA Controller 2311 typically contained within the Cable Modem 2300 transmits a synchronized DOCSIS CPE Controller clock 2312, to the coupled HPNA phone adapter 2400. The HPNA Phone Adapter 2411 includes a similar HPNA controller 2411 for extracting clock information transmitted utilizing conventional transmission protocols by the cable modem's HPNA controller 2311. Transmission is accomplished via clock transmission MAC messages link 2080. The HPNA Phone Adapter uses the clock information to frequency lock the HPNA Phone adapter internal clock 2412 to the DOCSIS CPE controller clock 2312.

Thus the timing distribution method causes the voice sampling circuit clock 2412 within the HPNA phone adapter to be frequency

locked to the DOCSIS CPE controller clock 2312. Also, the voice sampling circuits 2310 within the Cable Modem 2300 are phase locked to the DOCSIS CPE controller clock 2312. Thus, both voice sampling circuits 2311, 2411 are frequency synchronized to the station reference 1040. The voice-sampling circuits in the cable modem 2300 and the HPNA phone adapter 2400 are perforce frequency synchronized to the PSTN Network timing via the station reference 1040.

The method includes utilization of a clock distribution system in which no metallic connection is needed to distribute the clocks to achieve synchronization. A metallic connection exists between station reference 1040 and stratum 3 reference clock 2011 via link 1060 and to the PSTN gateway 1050 via link 1065. The metallic connections are well known and described in the previously mentioned TR-NWT-001244 specification.
Other than the previously mentioned metallic connections, the system distributes 2040, 2100, 2080 timing based upon timing messages that include clock information.

The PSTN Gateway 1050 performs a media conversion function where packet based voice data is received on a first interface form the Ethernet 2090 and converts the samples to a conventional PSTN sample based interface. Those skilled in the art will recognize that the PSTN interface can equivalently be any of a large variety of interface types. In the present embodiment, this interface is assumed to be a conventional T1 interface as described by Telcordia specification GR-303.
The T1 interface is a digital interface where samples are transmitted synchronously over a speed serial multiplexed interface. The PSTN gateway 1050 collects constant size sets of samples and constructs transmission packets that are transmitted via the available data transmission network to the connected target circuits. In this embodiment the target circuits are the Voice circuit 2310 contained in the Cable Modem 2300 or the HPNA Phone Adapter 2400. The present embodiment uses DOCSIS to transmit data over a Hybrid Fiber Coax (HFC) network 2060 and an Ethernet network 2090 to perform data packet delivery. Those skilled in the art will recognize that these are simple examples

of data transmission networks and that equivalently a large number of alternative network transmission systems are well known in this art to accomplish the same connections.

The Customer Premise Equipment (CPE) Voice sampling circuits receive the data packets containing the constant size set of voice samples and play these sample out to an audio interface to the connected telephone device 2001 using the frequency locked local version of the DOCSIS CPE controller clock 2312. This clock is frequency locked to the PSTN timing distribution clock via the Station reference 1040. Thus, these samples will play out at the same rate at the voice sampling circuit 2310, 2410 at the same rate that they are arriving at the PSTN gateway 1050. Hence the entire operation is free of data over run or under run impairments that tend to have an adverse affect on the voice quality that would tend to occur if this timing distribution method were not used.

Distribution is a DOCSIS transmission system accomplished by the following method. A conventional DOCSIS transmission system includes a DOCSIS head-end controller 2010, including CMTS master clock 2012. An HFC network 2060 is coupled to the DOCSIS head-in controller 2013 by messaging path 2050, 2040. The HFC network 2060 is coupled by messaging path 2040, 2050 to a DOCSIS CPE controller 2313. The DOCSIS CPE controller 2313 includes a local clock 2312. The local clock 2312 is synchronized to clock 2012 by a conventional internally generated DOCSIS clock sync method 2040. Clocks 2312 and 2012 are thus synchronized by a conventional DOCSIS mechanism. In the embodiment described in the DOCSIS system, clock 2012 is the master reference and establishes the time base for the entire DOCSIS network.

A conventionally formatted DOCSIS message includes a message called a sync message that transmits clock rate information concerning clock 2012 so that the controller 2313 contained in the cable modem 2300 uses that information to synchronize clock 2312 to clock 2012. This is the DOCSIS clock transport mechanism.

An embodiment of the invention utilizes the DOCSIS clock transport. The DOCSIS clock transport mechanism is designed solely to transmit a clock signal upstream. The DOCSIS clock

transport system is purely a transmitter clock for aiding internal DOCSIS network timing, i.e., the clock is neither imported nor exported.

The embodiment of the invention utilizes a stratum 3 reference clock 2011 to import a master clock. The stratum 3 reference clock 2011 synchronizes itself to the station reference clock 1040. A synchronization signal 2014 synchronizes the CMTS clock 2012 to the stratum reference clock 2011. The stratum 3 reference clock is conventionally constructed as outlined in Belcor standard TR 1244, the contents of which are incorporated in their entirety into this application by reference. Synchronization of a station reference 1040 to a Stratum reference clock 1060 is achieved by conventional synchronization circuitry known to those skilled in the art. Thus connected, the stratum 3 reference clock 2011 is now the CMTS 2010, CMTS master reference.

In the embodiment shown, the DOCSIS master reference 6 is slaved to the station reference 2112 through the stratum 3 reference clock 2011. When the DOCSIS system is operating, it transmits clock 2012 to clock 2312. However, what the DOCSIS system is actually doing is transmitting the station reference 1040, since clock 2012 is slaved to clock 1040 which is in turn slaved to the station reference 1040. It is desirable to slave the DOCSIS timing to the station reference 1040 that is also utilized by the PSTN network, since the PSTN is being interfaced to by the HPNA phone system. Thus in effect, the gateway 1050 is operating off of the station reference 1040.

The gateway converts packets arriving from the HFC 2060. The gateway 1050 converts packets arriving from the HFC via the Internet into a PSTN compatible signal. The entire PSTN network is synchronized by the station reference 1040. It is desirable to have packets of data arriving at the gateway 1050 to be timed in synchronization with the station reference 1040 to prevent slips.

The gateway 1050 is a computer that performs protocol conversions between different types of networks and applications allowing signals to be transferred. For example a gateway converts

messages between two differing protocols so that a message from a first network may be transported and processed in a second differing network. An Open System Interconnection (OSI) model defines a framework for implementing protocols in layers that is utilized in a gateway. Processing control is passed from one layer to the next, starting at the application layer at a station, and proceeding to the bottom protocol layer, over a channel (such as the internet) to a next station and back up a layered hierarchy at that station. Alternatively a message may be simply passed through a network, once its protocol is converted by a gateway so that it may pass through the network to a different network where it will be processed.

Data arrives at the gateway 1050 via an upstream path that orginates from one of several telephone sets 2002, 2001. The upstream data path for a HPNA phone to the PSTN starts with data path 2070 between the HPNA phone adapter and the cable modem 2300. The next link is from the cable modem to the HFC 2060 via link 2200. The next link is from the HFC to the CMTS 2010 via the upstream data path 2050. The CMTS links upstream data to the Internet via data path 3016. Finally, the Ethernet links the data to the gateway 1050 via data path 1070. At the gateway 1050, it is desirable to transfer the data to the PSTN 1000 without slips.

A slip free environment is alternatively termed a completely synchronous environment. By controlling a sample clock with external station reference 1040, the voice sampling circuit 2310, 2410, is completely synchronized to the station reference to provide a slip free conversion. Clock information is used to transmit and receive data. Clock information is also used to develop a sample clock to sample an audio interface at the gateway 1050. Audio samples are converted to data at the station reference rate of the PSTN. This synchronous sampling prevents slipping, simplifying circuitry recognition and tending to improve audio quality.

Slipping occurs when two clocks are not the same, such as the clock for the voice sampling circuit 2310 and the clock for the PSTN which is the station reference 1040. Often the clocks will be close but not the same. In a PSTN, network slip

management is utilized. For example, if the voice sampling clock were to be running slightly faster than the station reference clock 1040, then over time the voice sampling circuit 11 would be collecting more samples than the PSTN synchronized to the station reference 1040. Thus, more samples than are capable of being transmitted to the PSTN network are collected. This occurs because the gateway clock 1050 is not fully synchronized to the station reference 1040. A buffer circuit associated with the voice sampling circuit 2310 typically stores the samples. However, if the voice sampling circuit is sampling at a faster rate than the gateway, can clock the data into the PSTN, then a buffer circuit associated with the sampling circuit 2310 will fill up over time and samples will be discarded because they are more than can be processed. To prevent this problem, a slip buffer is typically utilized. In the slip buffer after a certain amount of time samples are discarded. After some of the information has been discarded, the buffer continues to fill up with data samples until a certain percentage of capacity has been reached when samples are again discarded.

In the case where the sampling clock of the voice sampling circuit 2310 is running slower than the station reference 1040 that is driving the synchronization circuitry in the gateway 1050, then the PSTN is accepting more data than the voice sampling circuit 2310 is capable of providing. To deal with this problem, the information is periodically repeated to maintain synchronization with the transmitter. The two techniques just outlined are often termed "slip buffer management". Thus, if the sampling clock 2310 is operating synchronously with the station reference 1040 that is clocking the gateway 1050, data will never slip. Data samples will be collected by the PSTN at exactly the same rate that they are being sent to the PSTN by the internet.

The timing synchronization in the downstream path is accomplished in the same way. Messages sent from the PSTN through the gateway 1050 are sampled with a clock set by the station reference 1040. The station reference is synchronized to a voice sampling circuit 2310 through the stratum 3 reference clock 2011 and the DOCSIS head-in controller through a message sent over the HFC to the DOCSIS CPE controller. This approach

to synchronization of clocks in a packet transport network allows management of slippage and the associated circuitry necessary to implement that slip management to be eliminated.

37110/RJP/B600

APPENDIX 1

CABLE MODEM SYSTEM WITH SAMPLE AND PACKET SYNCHRONIZATION

BACKGROUND

A desired solution for high speed data communications appears to be cable modem. Cable modems are capable of providing data rates as high as 56 Mbps, and is thus suitable for high speed file transfer, including applications such as bit-rate sampled data transmission to and from telephones, faxes or modem devices.

However, when transmitting packet based voice using cable modems, there is a need to synchronize voice packet sampling with cable modem system grant processing. The present invention provides a solution for such need.

DESCRIPTION OF THE DRAWINGS

FIG. 1 shows in simplified block diagram form an environment within which the present invention operates.

FIG. 2 shows in simplified block diagram form the interconnection of an exemplary home utilizing the present invention in accordance with a cable modem and cable modem termination system.

FIG. 3 shows in graphical form the allocation of time slots by the cable modem termination system.

FIGS. 4 and 5 shows in flow diagram form the construction of a frame.

FIGS. 6 and 7 show in simplified block diagram form a portion of the cable modem termination system which receives requests from the cable modems and which generates MAPS in response to the requests.

37110/RJP/B600

FIGS. 8 and 9 show in flow diagram form how a cable modem and cable modem termination system cooperate for packets transmitted by the cable modem to the cable modem termination system.

FIGS. 10 and 11 show in block diagram form aspects of the timing synchronization system between the cable modem and the cable modem termination system.

FIG. 12 shows in block diagram form an exemplary timing recovery circuit of a cable modem in more detail.

FIG. 13 shows in table form an example of coarse and fine coefficients suitable for various different update rates and bandwidths.

FIG. 14 shows in graphical form a timing slot offset between the cable modem clock and the cable modem termination system clock.

FIG. 15 shows in simplified block diagram form the burst transmission and reception by the cable modem and the cable modem termination system.

FIG. 16 shows the cable modem termination system in further detail.

FIGS. 17, 18 and 19 shows in graphical form relationships between grants and samples.

FIG. 20 shows in simplified block diagram form a representative embodiment of the present invention.

FIG. 21 shows in simplified block diagram form the operation of a headend clock synchronization circuit in accordance with the present invention.

FIG. 22 shows in simplified block diagram form the operation of a cable modem clock synchronization in accordance with the present invention.

FIGS. 23a, 23b and 23c show in graphical form the inter-relationship of signals used in accordance with the present invention.

37110/RJP/B600

FIGS. 24a, 24b and 24c show in graphical for the inter-relationship of further signals used in accordance with the present invention.

FIGS. 25, 26 and 27 show in simplified block diagram and graphical form grant time calculation circuitry in accordance with the present invention.

FIG. 28 shows in simplified block diagram form the inter-relationship between grant time circuitry, digital signal processor and buffers in accordance with the present invention.

FIGS. 29a and 29b shows in flow diagram form an operational DSP system software decision implementation in accordance with the present invention.

DETAILED DESCRIPTION

A description of the cable modem and cable modem termination system aspects in accordance with the present invention is first provided. A description of the voice sample and packet synchronization aspects in accordance with the present invention is then provided.

<u>Cable Modems and the Cable Modem Termination System</u>

In a cable modem system, a headend or cable modem termination system (CMTS) is located at cable company facility and functions as a modem which services a large number subscribers. Each subscriber has a cable modem (CM). Thus, the CMTS facilitates bidirectional communication with any desired one of the plurality of CMs.

The CMTS communicates with the plurality of CMs via a hybrid fiber coaxial (HFC) network, wherein optical fiber provides communication to a plurality of fiber nodes and each fiber node typically serves approximately 500 to 2,000 subscribers, which communicate with the node via coaxial cable. The hybrid fiber coaxial network of a CM system utilizes a point-to-multipoint

37110/RJP/B600

topology to facilitate communication between the CMTS and the plurality of CMs. Frequency domain multiple access (FDMA)/time division multiplexing (TDM) is used to facilitate communication from the CMTS to each of the CMs, i.e., in the downstream direction. FDMA /time domain multiple access (TDMA) is used to facilitate communication from each CM to the CMTS, i.e.,in the upstream direction.

The CMTS includes a downstream modulator for facilitating the transmission of data communications therefrom to the CMs and an upstream demodulator for facilitating the reception of data communications from the CMs. The downstream modulator of the CMTS utilizes either 64 QAM or 256 QAM in a frequency band of 54 MHz to 860 MHz to provide a data rate of up to 56 Mbps.

Similarly, each CM includes an upstream modulator for facilitating the transmission of data to the CMTS and a downstream demodulator for receiving data from the CMTS. The upstream modulator of each CM uses either QPSK or 16 QAM within the 5 MHz to 42 MHz bandwidth of the upstream demodulator and the downstream demodulator of each CM utilizes either 64 QAM or 256 QAM in the 54 MHz to 860 MHz bandwidth of the downstream modulator (in North America).

Referring now to FIG. 1, a hybrid fiber coaxial (HFC) network 1010 facilitates the transmission of data between a headend 1012, which includes at least one CMTS, and a plurality of homes 1014, each of which contains a CM. Such HFC networks are commonly utilized by cable providers to provide Internet access, cable television, pay-per-view and the like to subscribers.

Approximately 500 homes 1014 are in electrical communication with each node 1016, 1034 of the HFC network 1010, typically via coaxial cable 1029, 1030, 1031. Amplifiers 1015 facilitate the electrical connection of the more distant homes 1014 to the nodes 1016, 1034 by boosting the electrical signals so as to desirably

37110/RJP/B600

enhance the signal-to-noise ratio of such communications and by then transmitting the electrical signals over coaxial conductors 1030, 1031. Coaxial conductors 1029 electrically interconnect the homes 1014 with the coaxial conductors 1030, 1031, which extend between amplifiers 1015 and nodes 1016, 1034.

Each node 1016, 1034 is electrically connected to a hub 1022, 1024, typically via an optical fiber 1028, 1032. The hubs 1022, 1024 are in communication with the headend 1012, via optical fiber 1020, 1026. Each hub is typically capable of facilitating communication with approximately 20,000 homes 1014.

The optical fiber 1020, 1026 extending intermediate the headend 1012 and each hub 1022, 1024 defines a fiber ring which is typically capable of facilitating communication between approximately 100,000 homes 1014 and the headend 1012.

The headend 1012 may include video servers, satellite receivers, video modulators, telephone switches and/or Internet routers 1018, as well as the CMTS. The headend 1012 communicates via transmission line 1013, which may be a T1 or T2 line, with the Internet, other headends and/or any other desired device(s) or network.

Referring now to FIG. 2, a simplified block diagram shows the interconnection of the headend 1012 and an exemplary home 1014, wherein a CM 1046 communicates with a CMTS 1042, via HFC network 1010. Personal computer 1048, disposed within the home 1014, is connected via cable 1011 to the CM 1046. More particularly, with respect to the present invention, bit-rate sampled data transmission devices 1047a and 1047b, such as telephones, fax or modem units, are connected to sample and packet synchronization subsystem (described in more detail below) which, in turn, interfaces to CM 1046. CM 1046 communicates via coaxial cable 1017 with the HFC network 1044, which, in turn, communicates via optical fiber 1020 with CMTS 1042 of the headend 1012. Internet router 1040 facilitates communication between

37110/RJP/B600

the headend 1012 and the Internet or any other desired device or network, and in particular with respect to the present invention, to any end user system to which a call is being placed from home 1014, such as to a call recipient 2002 connected to the Public Switched Telephone Network (PSTN)through PSTN gateway 2004.

In order to accomplish TDMA for upstream communication, it is necessary to assign time slots within which CMs having a message to send to the CMTS are allowed to transmit. The assignment of such time slots is accomplished by providing a request contention area in the upstream data path within which the CMs are permitted to contend in order to place a message which requests additional time in the upstream data path for the transmission of their message. The CMTS responds to these requests by assigning time slots to the CMs making such a request, so that as many of the CMs as possible may transmit their messages to the CMTS utilizing TDMA and so that the transmissions are performed without undesirable collisions. In other words, the CM requests an amount of bandwidth on the cable system to transmit data. In turn, the CM receives a "grant" of an amount of bandwidth to transmit data in response to the request. This time slot assignment by the CMTS is known as a "grant" because the CMTS is granting a particular CM permission to use a specific period of time in the upstream.

Because of the use of TDMA, the CMTS uses a burst receiver, rather than a continuous receiver, to receive data packets from CMs via upstream communications. As those skilled in the art will appreciate, a continuous receiver can only be utilized where generally continuous communications (as opposed to burst communications as in the present invention) are performed, so as to substantially maintain timing synchronization between the transmitter and the receiver, as is necessary for proper reception of the communicated information. During continuous

37110/RJP/B600

communications, timing recovery is a more straightforward process since signal acquisition generally only occurs at the initiation of such communications. Thus, acquisition is generally only performed in continuous receivers once per continuous transmission and each continuous transmission may be very long.

However, the burst communications inherent to TDMA systems require periodic and frequent reacquisition of the signal. That is, during TDMA communications, the signal must be reacquired for each separate burst transmission being received.

The assignment of such time slots is accomplished by providing a request contention area in the upstream data path within which the CMs are permitted to contend in order to place a message which requests time in the upstream data path for the transmission of their message. The CMTS responds to these requests by assigning time slots to the CMs making such a request, so that as many of the CMs as possible may transmit their messages to the CMTS utilizing TDMA and so that the transmissions are performed without undesirable collisions.

Briefly, upstream data transmission on an upstream channel is initiated by a request made by a CM for a quantity of bandwidth, i.e., a plurality of time slots, to transmit data comprising a message. The size of the request includes payload, i.e., the data being transmitted, and overhead, such as preamble, FEC bits, guard band, etc. After the request is received at the headend, the CMTS grants bandwidth to the requesting CM and transmits the size of the grant and the specific time slots to which the data is assigned for insertion to the requesting CM.

It is important to understand that a plurality of such CMs are present in a CM system and that each of the CMs may, periodically, transmit a request for a time slot allocation to the CMTS. Thus, the CMTS frequently receives such requests and allocates time slots in response to such requests. Information representative of the allocated time slots is compiled to define

37110/RJP/B600

a MAP and the MAP is then broadcast to all of the CMs on a particular channel, so as to provide information to all of the CMs which have one or more data packets to transmit to the CMTS precisely when each of the CMs is authorized to transmit its data packets).

Referring now to FIG. 3, the allocation of time slots by the CMTS and the generation of a MAP which defines the time slot allocations is described in more detail. The contents of a MAP protocol data unit (PDU) 113 are shown. The MAP PDU 113, which is transmitted on the downstream channel by the CMTS 1042 to all of the CMs 1046 on a given frequency channel, contains the time slot allocations for at least some of the CMs 1046 which have previously sent a request to transmit one or more data packets to the CMTS 1042. When the channel bandwidth is sufficient, in light of the number of such requests received by the CMTS 1042, then the CMTS 1042 allocates a time slot for each such requesting CM 1046.

Further, the MAP PDU 113 at least occasionally defines at least one request contention region 112 and generally also contains a plurality of CM transmit opportunities 114 within the upstream channel 117. A maintenance frame 116 may also be defined by the MAP PDU 113 within the upstream channel 117, as discussed in detail below.

The request contention region 112 includes at least one time area within which the CMs 1046 transmit their requests to transmit data packets to the CMTS 1042. Each of the CM transmit opportunities 114 define a time slot within which a designated CM 1046 is permitted to transmit the data packet for which the request was previously sent to the CMTS 1042.

Additionally, one or more optional transmit contention regions (not shown) may be provided wherein CMs 1046 may contend for the opportunity to transmit data therein. Such transmit contention regions are provided when sufficient bandwidth is left

over after the MAP PDU 113 has allocated transmit opportunities 114 to all of those CMs 1046 which have requested a time slot allocation. Thus, transmit contention regions are generally provided when upstream data flow is comparatively light.

The upstream channel 119, is divided into a plurality of time intervals 110, each of which may optionally be further subdivided into a plurality of sub-intervals 115. The upstream channel 119 thus partitioned so as to facilitate the definition of time slots, such that each of a plurality of CMs 1046 may transmit data packets to the CMTS 1042 without interfering with one another, e.g., without having data collisions due to data packets being transmitted at the same time.

Thus, the use of a MAP 113 facilitates the definition of slots 92. Each slot 92 may be used for any desired predetermined purpose, e.g., as a request contention region 112 or a transmit opportunity 114. Each slot 92, as defined by a MAP PDU 113, includes a plurality of time intervals 110 and may additionally comprise one or more sub-intervals 115 in addition to the interval(s) 110. The number of intervals 110 and sub-intervals 115 contained within a slot 92 depends upon the contents of the MAP PDU 113 which defines the slot 92. The duration of each interval 110 and sub-interval 115 may be defined as desired. Optionally, each sub-interval 115 is approximately equal to a media access control (MAC) timing interval. Each MAP PDU 113 defines a frame and each frame defines a plurality of slots 92.

The beginning of each sub-interval 115 is aligned in time with the beginning of each interval 110 and each interval 110 typically contains an integral number of sub-intervals 115.

Typically, the request contention region 112 and each CM transmit opportunity 114 includes a plurality of integral time intervals 110. However, the request contention region 112 and/or the CM transmit opportunity 114 may alternatively include any desired combination of intervals 110 and sub-intervals 115.

37110/RJP/B600

Thus, each request contention region 112 may be utilized by a plurality of the CMs 1046 to request one or more time slot allocations which facilitate the transmission of one or more data packets during the CMs 1046 subsequently allocated transmit opportunity 114.

Each data packet may contain only data, although an extended data packet may be defined to include both data and a preamble. The preamble is typically stripped from an extended packet by the CMTS 1042 and the data in the packet is then processed by a central processing unit of the CMTS 1042.

The duration of the request contention region 112 is typically variable, such that it may be sized to accommodate the number of CMs 1046 expected to request time slot allocations from the CMTS 1042. The duration of the request contention region 112 may thus be determined by the number of requests transmitted by CMs as based upon prior experience.

The time slot allocations 92 defined by CM transmit opportunities 114 may optionally be defined, at least in part, on the basis of priorities established by the CMTS 1042 for different CMs 1046. For example, priorities may be established for individual CMs 1046 on the basis of an election made by the subscribers, which is typically dependent upon the type of service desired. Thus, a subscriber may elect to have either a premium (high priority) service or a regular (low priority) service.

Alternatively, priorities may be established by the CMTS 1042 for the CMs based upon size and number of CM transmit opportunities 114 historically requested by the subscribers. Thus, a CM that typically requires a large number of time intervals 110 may be defined as a high priority user, and thus given priority in the allocation of time slots within a CM transmit opportunity 114, based upon the assumption that such large usage is indicative of a continuing need for such priority,

e.g., is indicative that the subscriber is utilizing cable television, pay-per-view or the like.

Alternatively, the CMTS may assign such priorities based upon the type of service being provided to each CM. Thus, for example, when cable television or pay-per-view is being provided to a CM, then the priority of that CM may be increased, so as to assure uninterrupted viewing.

The priority associated with each CM 1046 may determine both the size of time slots allocated thereto and the order in which such allocations are performed. Those allocations performed earlier in the allocation process are more likely to be completely filled than those allocations performed later in the allocation process. Indeed, allocations performed later in the allocation process may go unfilled, when the bandwidth of the channel is not sufficient to facilitate allocation of time slots for all requesting CMs 1046.

Time slots which define the maintenance region 116 are optionally provided in a MAP 113. Such maintenance regions 116 may be utilized, for example, to facilitate the synchronization of the clocks of the CMs with the clock of the CMTS. Such synchronization is necessary in order to assure that each CM 1046 transmits only within its allocated time slots, as defined by each CM's transmit opportunity 114.

The request contention region 112 CM transmit opportunity 114 and maintenance region 116 typically begin at the beginning of an interval 110 and end at the end of an interval 110. However, each request contention region 112, CM transmit opportunity 114 and maintenance region 116, may begin and end anywhere as desired. Thus, variable duration request contention regions 112, CM transmit opportunities 114 and maintenance regions 116 are provided. Such variable duration request contention regions 112, transmit opportunities 114 and maintenance regions 116 facilitate flexible operation of the CM

37110/RJP/B600

system and enhance the efficiency of data communications on the CM system by tending to mitigate wasted channel capacity.

The current MAP 170 is transmitted in the downstream channel 111 after transmission of a previous MAP 90 and before any subsequent MAPs 91. Data, such as data packets associated with web pages, e-mail, cable television, pay-per-view television, digital telephony, etc. are transmitted between adjacent MAPs 90, 170, 91.

The contents of each CM transmit opportunity 114 optionally include data and a preamble. The data includes at least a portion of the data packet for which a request to transmit was sent to the CMTS 1042. The preamble typically contains information representative of the identification of the CM 1046 from which the data was transmitted, as well as any other desired information.

The data and the preamble do not have to occupy the full time interval of the cable transmit opportunity 114. Guard bands are optionally provided at the beginning and end of each slot, so as to decrease the precision with which time synchronization between the CMTS and each CM must be performed. Thus, by providing such guard bands, some leeway is provided in the transmit time during which each CM inserts its data packet into the upstream channel 119.

Referring now to FIGS. 4 and 5, the construction of a frame is shown. As shown in block 143, requests are made by the CMs 1046 in a request contention region 112 of a first MAP for the grant or allocation by the CMTS 1042 to the subscribers of Information Elements (IE). An Information Element may be considered to be the same as a region. A maintenance opportunity is optionally provided as shown at block 144. Such maintenance opportunities may, for example, be used to synchronize the operation of the CM 1046 with the operation of the CMTS 1042.

37110/RJP/B600

As previously indicated, this maintenance opportunity may be provided only periodically.

A determination is then made at block 146 as to whether the high priority request queue is empty. If the answer is "No" with respect to the high priority request queue, a determination is then made at block 148 as to whether the frame length is less than a desired length. If the answer is "Yes", the request of the subscriber to transmit data is granted and the frame length is incremented by the size of the data requested at block 150.

If the high priority request queue is empty, a determination is made at block 152 as to whether the low priority request queue is empty. If the answer is "No", a determination is made at block 154 as to whether the frame length will be less than the desired length. If the answer is "Yes" with respect to the low priority request queue, the request of the CM 1046 to transmit data to the CMTS 1042 is granted and the frame length is incremented by the size of the grant. This is indicated at block 156.

It may sometimes happen that the frame length will be at least equal to the desired length when the request with respect to the high priority request queue is introduced to the block 148. Under such circumstances, the request is not granted and a determination is then made as to whether the low priority request queue is empty. Similarly, if the frame length will be greater than the desired frame length when a request with respect to the low priority request queue is made, the request is not granted. An indication is accordingly provided on a line 157 when the high priority request queue and the low priority request queue are both empty or when the frame length will be at least as great as the desired length.

When the high priority request queue and the low priority request queue are both empty or when the frame length will be at least as great as the desired length upon the assumed grant of

37110/RJP/B600

a request, a determination is made, as at block 158 (FIG. 7) as to whether the request queues are empty. This constitutes an additional check to make sure that the queues are empty. If the answer to such determination is "No", this indicates that the frame length will be greater than the desired frame length upon the assumed grant of a request. Under such circumstances, a grant of a zero length is provided in the MAP 170 for each request in each queue. This zero length grant is provided so that the headend can notify the subscriber that the request has not been granted but was received by the headend. In effect, a zero length grant constitutes a deferral. The request was seen, i.e., not collided, but not granted yet. It will be granted in a subsequent MAP 91.

If a determination is made as at block 158 that the request queues are empty, a determination is then made at block 162 as to whether the frame length will be less than the desired frame length. If the answer is "Yes", the frame is padded to the desired length with data from a contention data region 168 in the frame, as indicated at block 164. The contention data region 168 constitutes an area of reduced priority in the frame. It provides for the transmission of data from the CMs 1046 to the CMTS 1042 via available slots in the frame where CMs have not been previously assigned slots by the CMTS 1042. The contention data region does not require a grant by the CMTS 1042 of a request from a CM 1046 as in the request contention data region 112 in FIG. 3. Since no grant from the CMTS 1042 is required, the contention data region 168 in FIG. 7 (described below in additional detail) provides faster access to data for the subscriber than the request contention region 112.

Available slots in a frame are those that have not been assigned on the basis of requests from the CMs 1046. As indicated at block 166 in FIG. 5, the CMTS 1042 acknowledges to the CM 1046 that the CMTS 1042 has received data from the

contention data region in the frame. The CMTS 1042 provides this acknowledgment because the CM 1046 would not otherwise know that such data was not involved in a data collision and has, indeed, has been received from the contention data region 168.

Referring now to FIGS. 6 and 7, a block diagram of that portion of the CMTS 1042 which receives requests from the CMs 1046 and which generator MAPs in response to those requests is shown. The contention data region 168 in FIG. 7 is included in frame 118 defined by a MAP 111 (FIG. 3). The frame 118 in FIG. 7 may include a number of other regions. One region is indicated at 172 and is designated as contention requests region 112 in FIG. 3. It includes slots designated as X 181. In these slots X 181, collisions between request data from different CMs 1046 have occurred. Other slots in the contention request region 172 are designated as R 183. Valid uncollided request data is present in these slots. The contention request region 172 also illustratively includes an empty slot 175. None of the subscribers 14 has made a request in this empty slot 175.

A CM transmit opportunity region 176 (corresponding to the CM transmit opportunity region 114 in FIG. 3) may also be provided in the frame 118 adjacent the contention request area 172. As previously indicated, individual CMs 1046 are assigned slots in this area for data in accordance with their requests and with the priorities given by the CMTS 1042 to these requests. Optionally, the CM transmit opportunity region 176 may be considered as having two sub-regions. In a sub-region 178, slots are specified for individual subscribers on the basis of requests of a high priority. Slots are specified in an area 180 for individual subscribers on the basis of requests of a low priority.

The frame 118 may optionally also include a maintenance region 182. This corresponds to the maintenance region 116 in FIG. 3. As previously described, the region 182 provides for a

37110/RJP/B600

time coordination in the clock signals of the CMTS 1042 and the CMs 1046. The frame 118 additionally may optionally include a region 184 in the contention data region 168 where a collision has occurred. Valid data is provided in an area 186 in the frame where no collision occurred. A blank or empty area 188 may exist at the end of the contention data region 186 where further data could be inserted, subject to potential collisions. It will be appreciated that the different regions in the frame 118, and the sequence of these different regions, are illustrative only and that different regions and different sequences of regions may alternatively be provided.

The signals of the frames 118 from different CMs 1046a, 1046b, 1046c, 1046d, etc. (FIG. 7) are introduced in upstream data processing through a common line 191 (FIGS. 6 and 7) to a TDMA demultiplexer 192 (FIG. 6) in the CMTS 1042. After demultiplexing, data in from the CMs 1046a, 1046b, 1046c, 1046d, etc. pass from the demultiplexer 192 to a data interface 194. The signals at the data interface 194 are processed in an Ethernet system (not shown) or the like. The operation of the MAP generator 198 is controlled by data requests from the individual CMs 1046a, 1046b, 1046c, 1046d, etc. and by collision information which is indicative of the CMs 1046a, 1046b, 1046c, 1046d, etc. attempts to insert data in the contention data region 168. Thus, for example, a large number of collision may indicate a need for a larger contention request region 172 in the next MAP. Attempts to insert data in the contention data region 168 may, optionally, be utilized by the MAP generator 198 to increase the priority of any CM unsuccessfully attempting to transmit such data. The MAPs generated by the MAP generator 198 pass through the multiplexer 196 and are broadcast by the CMTS 1042 to the CMs 1046a, 1046b, 1046c, 1046d.

A sample MAP generated by the MAP generator 198 is generally indicated at 202 in FIG. 6. The MAP 202 includes a region 204

37110/RJP/B600

where the requests of the CMs 1046 for Information Elements (IE) within which to transmit data are indicated. As previously indicated, an Information Element (IE) may be considered to be the same as a region. The MAP 202 also includes a region 206 where the CMTS 1042 has granted the requests of the subscribers for Information Elements to transmit data. The MAP 202 additionally includes a contention data region 208 where the CMTS 1042 has given the CMs 1046 the opportunity to transmit data in available spaces or slots without specifying the open spaces or slots where such transmission is to take place. An acknowledgment region 210 is also included in the MAP 202. In this region, the CMTS 1042 acknowledges to the CM 1046 that it has received data from the subscribers in the available slots in the contention data region 208. As discussed above, the CMTS 1042 has to provide such acknowledgment because the CMs 1046 will not otherwise know that the CMTS 1042 has received the data from the CMs 1046 in the contention data region 208.

FIGS. 8 and 9 define a flowchart, generally indicated at 600, in block form and show how the CM 1046 and the CMTS 1042 cooperate for packets transmitted by the CM 1046 to the CMTS 1042. The operation of the blocks in the flowchart 600 is initiated at a start block 602. As indicated at block 604 in FIG. 8, the CM 1046 then awaits a packet from an external source. For example, the external source may be a personal computer (PC) 1048, or bit-rate sampled data transmission device 1047a, 1047b (FIG. 2) at the home 1014 of a subscriber. As shown in block 606, the CM 1046 then submits to the CMTS 1042 a bandwidth request for enough time slots to transmit the packet. Upon receipt of the request, the CMTS sends a grant or partial grant to the CM in the MAP. The CM 1046 then checks at block 610 to determine if the CMTS 1042 has granted the request, or any portion of the request, from the CM 1046. In block 610, SID is an abbreviation of Service Identification, for example, a SID

37110/RJP/B600

assigned to bit-rate sampled data transmission device 1047a. If the answer is "Yes" (see line 611 in FIGS. 8 and 9), the CM 1046 then determines if the CMTS 1042 has granted the full request from the CM 1046 for the bandwidth. This corresponds to the transmission of the complete data packet from the CM 1046 to the CMTS 1042. This is indicated at block 612 in FIG. 9.

If the answer is "Yes", as indicated at block 614 in FIG. 9, the CM 1046 determines if there is another packet in a queue which is provided to store other packets awaiting transmission to the CMTS 1042 from the CM 1046. This determination is made at block 616 in FIG. 8. If there are no other packets queued, as indicated on a line 617 in FIGS. 8 and 9, the CM 1046 sends the packet without a piggyback request to the CMTS 1042 (see block 618 in FIG. 8) and awaits the arrival of the next packet from the external source as indicated at 604. If there are additional packets queued as indicated by a line 619 in FIGS. 8 and 9, the CM 1046 sends to the CMTS 1042 the packet received from the external source and piggybacks on this transmitted packet a request for the next packet in the queue. This is indicated at 620 in FIG. 10. The CM then returns to processing MAPs at 608 looking for additional grants. The CMTS 1042 then processes the next request from the CM.

The CMTS 1042 may not grant the full request for bandwidth from the CM 1046 in the first MAP 111. The CMTS 1042 then provides this partial grant to the CM 1046. If the CMTS operates in multiple grant mode, it will place a grant pending or another grant in the MAP in addition to the partial grant it sends to the CM. The CM processes the MAPs as shown in 608 and sees the grant in 611. The grant is smaller than the request as on 622 so the CM calculates the amount of the packet that will fit in the grant as on 624. With a multiple grant mode CMTS, the CM will see the partial grant with an additional grant or grant pending in subsequent MAPs as in 610 and 611. The CM then sends the

37110/RJP/B600

fragment, without any piggyback request as in 628 and 630 to the CMTS 1042. The CM returns to processing MAP information elements in 608 until it gets to the next grant. The CM then repeats the process of checking to see if the grant is large enough as in 612. If the next grant is not large enough, the CM repeats the process of fragmenting the remaining packet data and, as in 626, checking to see if it needs to send a piggyback request based on additional grants or grant pendings in the MAP. If the grant is large enough to transmit the rest of the packet as on 614, the CM checks to see if there is another packet enqueued for this same SID. If so, the CM sends the remaining portion of the packet with the fragmentation header containing a piggyback request for the amount of time slots needed to transmit the next packet in the queue as on line 620. The CM then returns to processing the MAP information elements. If there is not another packet enqueued for this SID, then the CM sends the remaining portion of the packet with fragmentation header containing no piggyback request as shown in 618. The CM then returns to 604 to await the arrival of another packet for transmission. When the CMTS 1042 partially grants the request from the CM 1046 in the first MAP 11 and fails to provide an additional grant or grant pending to the CM 1046 in the first MAP, the CM will not detect additional grants or grant pendings as on 632. The CM 1046 then sends to the CMTS 1042 a fragment of the data packet and a piggyback request for the remainder as in 634. When the CM has transmitted the fragment with the piggybacked request as shown on line 638, the CM returns to processing MAP information elements as in 608 while waiting for additional grants. When the CMTS receives the fragment with the piggybacked request, the CMTS must decide whether to grant the new request or send a partial grant based on the new request. This decision is based on the scheduling algorithms implemented on the CMTS.

37110/RJP/B600

Any time during the request/grant process, the CMTS could fail to receive a request or the CM could fail to receive a grant for a variety of reasons. As a fail safe mechanism, the CMTS places an acknowledgment time, or ACK time, in the MAPs it transmits. This ACK time reflects the time of the last request it has processed for the current MAP. The CM uses this ACK time to determine if its request has been lost. The ACK timer is said to have "expired" when the CM is waiting for a grant and receives a MAP with an ACK time later in time than when the CM transmitted its request. As the CM is looking for grants at 610, if the ACK time has not expired as on 644, the CM returns to processing the MAPs as in 608. If the ACK timer does expire as on 646, the CM checks to see how many times it has retried sending the request in 648. If the number of retries is above some threshold, the retries have been exhausted as on 654 and the CM tosses any untransmitted portion of the packet at 656 and awaits the arrival of the next packet. If the ACK timer has expired and the number of retries have not been exhausted as in 650, the CM uses a contention request region to transmit another request for the amount of time slots necessary to transmit the untransmitted portion of the packet as in 652. The CM then returns to processing the MAPS.

Referring to FIG. 10, the CMTS 1042 includes a crystal oscillator timing reference 16 which provides an output to a headend clock synchronization circuitry 18. It is this timing reference 16 to which each of the CMs 1046 must be synchronized. Headclock clock synchronization circuitry also receives an input from network clock reference 2003, which will be discussed in more detail below. The headend clock synchronization circuit 18 is incremented by the output of the crystal oscillator timing reference 16 and maintains a count representative of the number of cycles provided by the crystal oscillator timing reference 16 since the headend clock synchronization circuit 18 was last

37110/RJP/B600

reset. The headend clock synchronization circuit 18 includes a free-running counter having a sufficient count capacity to count for several minutes before resetting.

A timebase message generator 20 receives the count of the headend clock synchronization circuit 18 to provide an absolute time reference 21 which is inserted into the downstream information flow 22 provided by downstream data queue 24, as discussed in detail below. The timebase message generator 20 prefers a module function, i.e., a saw tooth pattern as a function of time) and the counter clock is generated by the oscillator with very tight accuracy.

Timing offset generator 26 receives a ranging signal message 27 from each individual CM 1046 with which the CMTS is in communication. The slot timing offset generator 26 provides a slot timing offset 28 which is representative of a slot timing offset between the CMTS 1042 and the CM 1046 and inserts the slot timing offset 28 into the downstream information flow 22. The slot timing offset 28 is calculated by determining the position of the slot timing offset from the expected time 27 within a dedicated timing slot of the upstream communications, as discussed in detail below. The timing effort generator 26 encodes the timing offset (ranging error) detected by the upstream receiver into a slot timing offset message. Slot timing offset messages are sent only after the frequency of the local reference clock has been acquired by the CM.

Downstream modulator 30 primarily modulates the downstream information flow 22. Absolute time references 21 are inserted at quasi-periodic intervals as determined by a timestamp send counter. A slot timing offset message 28 is inserted after measuring the slot timing error upon the arrival of a ranging signal message 27.

37110/RJP/B600

The time line 32 of the CMTS 1042 shows that the slot timing offset 28 is the difference between the expected receive time and the actual receive time of the slot timing offset message 27.

Each CM 1046 includes a downstream receiver 34 for facilitating demodulation of the data and timestamp message, and timing recovery of downstream communications from the CMTS 1042. The output of the downstream receiver 34 is provided to timebase message detector 36 and slot timing offset detector 38. The downstream information (any data communication, such as a file transfer or MPEG video signal) received by the downstream receiver 34 is also available for further processing, as desired.

The timebase-message detector 36 detects the timebase message generated by timebase message generator 20 of the CMTS 1042. Similarly, the slot timing offset detector 38 detects the slot timing offset 28 generated by the slot timing offset generator 26 of the CMTS 1042. The timebase message detector 36 provides an absolute time reference 40 which is representative of the frequency of the crystal oscillator timing reference 16 of the CMTS 1042. The absolute time reference 40 is provided to a digital tracking loop 42 which provides a substantially stable clock output for the CM 1046 which corresponds closely in frequency to the frequency of the crystal oscillator timing reference 16 of the CMTS 1042. Thus, the digital tracking loop 42 uses the absolute time reference 40, which is representative of the frequency of the crystal oscillator timing reference 16, to form an oscillator drive signal which drives a numerically controlled oscillator 44 in a manner which closely matches the frequency of the crystal oscillator timing reference 16 of the CMTS 1042, as discussed in detail below.

A difference between the absolute time reference 40 and the output of a local time reference 46, which is derived from the numerically controlled oscillator 44, is formed by differencing circuit 48. This difference defines a frequency error value

37110/RJP/B600

which represents the difference between the clock of the CM 1046 (which is provided by local time reference 46) and the clock of the CMTS 1042 (which is provided by crystal oscillator timing reference 16).

This frequency error value is filtered by loop averaging filter 50 which prevents undesirable deviations in the frequency error value from affecting the numerically controlled oscillator 44 in a manner which would decrease the stability thereof or cause the numerically controlled oscillator 44 to operate at other than the desired frequency. The loop filter 50 is configured so as to facilitate the rapid acquisition of the frequency error value, despite the frequency error value being large, and then to reject comparatively large frequency error values as the digital tracking loop 42 converges, i.e., as the output of the local timing reference 46 becomes nearly equal to the absolute time reference 40, thereby causing the frequency error value to approach zero.

An initial slot timing offset 52 is added by summer 54 to the output of the local time reference 46 to provide a partially slot timing offset corrected output 56. The partially slot timing offset corrected output 56 of summer 54 is then added to slot timing offset 58 provided by slot timing offset detector 38 to provide slot timing offset and frequency corrected time reference 60. The timing offset correction block is a simple adder which adds two message values. Such simplified operation is facilitated only when the resolution of the timing offset message is equal to or finer than that of the timestamp message.

The initial slot timing offset 52 is merely an approximation of the expected slot timing offset likely to occur due to the propagation and processing delays, whose approximate values have been predetermined. After frequency conversion using the phase locked loop and timebase message error, the slot timing offset 58 provides a final correction which is calculated by the CMTS

37110/RJP/B600

1042 in response to the CMTS 1042 receiving communications from the CM 1046 which are not properly centered within their desired timing slots, as discussed in detail below.

Scaler 62 scales the frequency corrected time reference 60 so as to drive upstream transmitter 69 at the desired slot timing.

Time reference 64 is compared to the designated transmit time 66 which was allocated via downstream communication from the CMTS 1042 to the CM 1046. When the time reference 64 is equal (at point 67) to the designated transmit time, then an initiate burst command 68 is issued and the upstream data queue 70 is modulated to form upstream transmission 72.

The timing offset (error) message is generated by the CMTS. The timing offset (error) is simply the difference between the expected time and the actual arrival time of the ranging message at the CMTS burst receiver.

Still referring to FIG. 10, although only one CM 1046 is shown in FIG. 10 for clarity, the CMTS 1042 actually communicates bidirectionally with a plurality of such CMs 12. Such communication as discussed herein may actually occur between the CM system and the plurality of CMs by communicating simultaneously with the CMs on a plurality of separate frequency channels. The present invention addresses communication of a plurality of different CMs on a single frequency channel in a serial or time division multiplexing fashion, wherein the plurality of CMs communicate with the CMTS sequentially. However, it will be appreciated that while this plurality of CMs is communicating on one channel with the CMTS (using time division multiple access or TDMA), many other CMs may be simultaneously communicating with the same CMTS on a plurality of different channels (using frequency division multiplexing/time division multiple access or FDM/TDMA).

37110/RJP/B600

Referring now to FIG. 11, the CMTS 1042 and the CM 1046 are described in further detail. The multiplexer 29 of the CMTS 1042 combines downstream information flow 22 with slot timing offset 28 from slot timing offset generator 26 and with absolute time reference 21 from timebase message generator 20 to provide downstream communications to the downstream transmitter, which includes downstream modulator 30 (FIG. 10). The slot timing offset generator 26 receives a slot timing offset signal 28 from the upstream receiver 25. The location of the slot timing offset signal within a timing slot of an upstream communication defines the need, if any, to perform a slot timing offset correction. Generally, a slot timing offset value will be transmitted, even if the actual slot timing offset is 0. When the slot timing offset message is desirably located within the timing offset slot, and does not extend into guard bands which are located at either end of the timing offset slot, then no slot timing offset correction is necessary.

However, when the slot timing offset message extends into one of the guard bands of the timing offset slot of the upstream communication, then a slot timing offset 28 is generated by the slot timing offset generator 26, which is transmitted downstream to the CM 1046 where the slot timing offset 28 effects a desired correction to the time at which upstream communications occur, so as to cause the slot timing offset message and other transmitted data to be positioned properly within their upstream data slots.

The headend tick clock 15 includes the crystal reference 16 of FIG. 10 and provides a clock signal to linear counting sequence generator 18. Slot/frame time generator 19 uses a clock signal provided by headend clock synchronization circuit 18 to provide both an minislot clock 21 and a receive now signal 23. The upstream message clock 21 is the clock by which the message slots are synchronized to effect time division multiple access

1 37110/RJP/B600

(TDMA) communications from each CM 1046 to the CMTS 1042. A Transmit now signal is generated at the beginning of each minislot of a transmission. A Receive now signal is similarly generated at the beginning of a received packet.

A minislot is a basic medium access control (MAC) timing unit which is utilized for allocation and granting of time division multiple access (TDMA) slots. Each minislot may, for example, be derived from the medium access control clock, such that the minislot begins and ends upon a rising edge of the medium access control clock. Generally, a plurality of symbols define a minislot and a plurality of minislots define a time division multiple access slot.

The CM 1046 receives downstream data from the downstream channel 14B. A timebase message detector 36 detects the presence of a timebase message 21 in the downstream data.

Slot timing offset correction 47 is applied to upstream communications 14A prior to transmission thereof from the subscriber CM 1046. The slot timing offset correction is merely the difference between the actual slot timing offset and the desired slot timing offset. Thus, the slot timing offset correction is generated merely by subtracting the actual slot timing offset from the desired offset. Slot/frame timing generator 49 transmits the upstream data queue 70 (FIG. 10) at the designated transmit time 66 (FIG. 10).

Summer 48 subtracts from the timebase message 21 of the local time reference 46 and provides an output to a loop filter 50 which drives numerically controlled oscillator 44, as discussed in detail below.

Upstream transmitter 11 facilitates the transmission of upstream communications 14A from the subscriber CM 1046A and upstream receiver 13A facilitates the reception of the upstream communications 14A by the CMTS 10.

37110/RJP/B600

Downstream transmitter 17 facilitates the transmission of downstream communications 14 from the CMTS 16 to the CM 1046 where downstream receiver 15 facilitates reception thereof.

Referring now to FIG. 12, an exemplary timing recovery circuit of a CM is shown in further detail. Downstream demodulator 95, which forms a portion of downstream receiver 15 of FIG. 11, provides clock and data signals which are derived from downstream communications 14B (FIG. 11). The data signals include downstream bytes which in turn include the count or timestamp 97 and timebase message header 81 transmitted by the CMTS 1042. Slot timing offset messages are included in the downstream flow of downstream data.

Timestamp detector 80 detects the presence of a timestamp header 81 among the downstream bytes and provides a timestamp arrived signal 82 which functions as a downstream byte clock sync. The timestamp arrived signal 82 is provided to synchronizer 83 which includes register 101, register 102, AND gate 103, inverter 104 and latch 105. Synchronizer 103 synchronizes the timestamp arrived signal 82 to the clock of the CM 1046, to provide a data path enable tick clock sync 107 for enabling the digital tracking loop 42.

When the digital tracking loop 42 is enabled by the data path enable tick clock sync 107 output from the synchronizer 83 in response to detecting a timestamp header by timestamp detector 80, then the timestamp, which is a count provided by the headend clock synchronization circuit 18 of FIG. 11, is provided to the digital tracking loop 42 and the digital tracking loop 42 is enabled so as to process the timestamp.

A differencing circuit or saturating frequency detector 109 compares the timestamp to a count provided to the saturating frequency detector 109 by timebase counter 111 which is representative of the frequency of numerically controlled oscillator 44. The saturating frequency detector 109 provides

37110/RJP/B600

a difference signal or frequency error value 112 which is proportional to the difference between the frequency of the numerically controlled oscillator 44 of the CM and the crystal oscillator reference 16 of the CMTS.

If the difference between the value of the timestamp and the count of timebase counter 111 is too large, indicating that the timestamp may be providing an erroneous value, then the saturating frequency detector 109 saturates and does not provide an output representative of the difference between the value of the timestamp and the count of timebase counter 111. In this manner, erroneous timestamps are not accepted by the digital tracking loop 42.

Pass 113 loop enable allows the difference provided by the saturating frequency detector 109 to be provided to latch 115 when a global enable is provided thereto. The global enable is provided to zero or pass 113 when functioning of the digital tracking loop 42 is desired.

Latch 115 provides the frequency error value 112 to a loop filter which includes multipliers 117 and 119, scalers 121 and 123, summers 124, 125 and latch 127.

The multipliers 117 and 119 include shift registers which effect multiplication by shifting a desired number of bits in either direction. Scalers 121 and 123 operate in a similar manner.

The loop filter functions according to well-known principles to filter out undesirable frequency error values, such that they do not adversely affect the stability or operation of numerically controlled oscillator 44. Thus, the loop filter tends to smooth out undesirable deviations in the frequency error value signal, so as to provide a more stable drive signal for the numerically controlled oscillator 44.

The multipliers 117 and 119 can be loaded with different coefficients such that the bandwidth of the loop filter may be

changed from a larger bandwidth during initial acquisition to a smaller bandwidth during operation. The larger bandwidth used initially facilitates fast acquisition by allowing frequency error values having larger deviations to be accepted. As the digital tracking loop 42 converges, the frequency error value tends to become smaller. At this time, frequency error values having larger deviations would tend to decrease stability of the digital tracking loop 42 and are thus undesirable. Therefore, different coefficients, which decrease the bandwidth of the loop filter, are utilized so as to maintain stability of the digital tracking loop 42.

A table showing an example of coarse and fine coefficients K0 and K1 which are suitable for various different update rates and bandwidths are shown in FIG. 13.

The output of the loop filter is provided to latch 131. The output of latch 131 is added to a nominal frequency by summer 133 so as to define a drive signal for numerically controlled oscillator 44.

Those skilled in the art will appreciate that the addition of a frequency offset, if properly programmed to a normal frequency, will decrease the loop's acquisition time. This is due to the fact that the final value of the accumulator 127 will be closer to its initial value.

The nominal frequency is generally selected such that it is close in value to the desired output of the numerically controlled oscillator 44. Thus, when the numerically controlled oscillator 44 is operating at the desired frequency, the filtered frequency error value provided by latch 131 is nominally zero.

Referring now to FIG. 14, a slot timing offset between the clock of the CM 1046 and the clock of the CMTS 1042 must be determined so as to assure that messages transmitted by the CM 1046 are transmitted during time slots allocated by the CM system 10. As those skilled in the art will appreciate, propagation

37110/RJP/B600

delays 400 and processing delays 402 combine to cause the CM 1046 to actually transmit at a later point in time than when it is requested to do so by the CMTS 1042. Thus, a slot timing offset must be provided to each CM 1046, to assure that it transmits at the correct time. This slot timing offset is determined by the CMTS 1042 by having the CMTS 1042 monitor a dedicated slot timing offset slot in upstream communications so as to determine the position of a slot timing offset message therein. The position of the slot timing offset message within the dedicated slot timing offset slot in the upstream communication determines the slot timing offset between the clock of the CMTS 1042 and the clock of the CM 1046. Thus, the CMTS 1042 may use this error to cause the CM 1046 to transmit at an earlier point in time so as to compensate for propagation and processing delays. This slot timing offset correction is equal to 2Tpg + Tprocess.

Initially, the slot timing offset slot includes a comparatively large time slot, i.e., having comparatively large guard times, so as to accommodate comparatively large slot timing offset error. In a normal data packet, the width of the timing offset slot may be reduced when slot timing offset errors become lower (thus requiring smaller guard bands), so as to *facilitate* more efficient upstream communications.

Generally, communications will be initialized utilizing a comparatively large guard time. After acquisition, when slot timing accuracy has been enhanced, then the guard time may be reduced substantially, so as to provide a corresponding increase in channel utilization efficiency.

According to a further aspect of the present invention, data packets are acquired rapidly, e.g., in an order of sixteen symbol or so, so as to facilitate enhanced efficiency of bandwidth usage. As those skilled in the art will appreciate, it is desirable to acquire data packets as fast as possible, so as to minimize the length of a header, preamble or other

non-information bearing portion of the data packet which is used exclusively for such acquisition.

As used herein, acquisition is defined to include the modifications or adjustments made to a receiver so that the receiver can properly interpret the information content of data packets transmitted thereto. Any time spent acquiring a data packet detracts from the time available to transmit information within the data packet (because of the finite bandwidth of the channel), and is therefore considered undesirable.

Acquisition includes the performance of fine adjustments to the parameters which are defined or adjusted during the ranging processes. During the ranging processes, slot timing, carrier frequency, and gross amplitude (power) of the data packet are determined. During acquisition, these parameters are fine-tuned so as to accommodate fractional symbol timing, carrier phase correction and fine amplitude of the data packet.

Moreover, a ranging process is used to control power, slot timing and carrier frequency in the upstream TDMA channel. Power must be controlled so as to provide normalized received power at the CMTS, in order to mitigate inter-channel interference. The carrier frequency must be controlled so as to ensure proper channelization in the frequency domain. Slot timing must be controlled so as to mitigate the undesirable collision of data packets in the time domain and to account for differential propagation delays among different CMs.

Referring now to FIG. 15, the CMTS 1042 comprises a burst receiver 292 for receiving data packets in the upstream data flow, a continuous transmitter 290 for broadcasting to the CMs 1046 via the downstream data flow and a medium access control (MAC) 60 for providing an interface between the burst receiver 292, the continuous transmitter 290 and other headend communications devices such as video servers, satellite

37110/RJP/B600

receivers, video modulators, telephone switches and Internet routers 1018 (FIG. 2).

Each CM 46 (FIG. 2) comprises a burst transmitter 294 for transmitting data to the CMTS 1042 via upstream data flow, a continuous receiver 296 for receiving transmissions from the CMTS 1042 via the downstream data flow and medium access control (MAC) 90 for providing an interface between the burst transmitter 294, the continuous receiver 296 and subscriber communications equipment such as a PC 48 (FIG. 2), a telephone, a television, etc.

The burst receiver 292, medium access control (MAC) 60 and continuous transmitter 290 of the CMTS 1042 and the burst transmitter 294, medium access control (MAC) 90 and continuous receiver 296 of each CM may each be defined by a single separate, integrated circuit chip.

Referring now to FIG. 16, the CMTS 1042 of FIG. 2 is shown in further detail. The CMTS 1042 is configured to receive signals from and transmit signals to an optical fiber 79 of the HFC network 1010 (FIG. 2) via optical-to-coax stage 49, which is typically disposed externally with respect to the CMTS 1042. The optical-to-coax stage 49 provides an output to the 5-42 MHz RF input 56 via coaxial cable 54 and similarly receives a signal from the RF up converter 78 via coaxial cable 52.

The output of the RF input 56 is provided to splitter 57 of the CMTS 1042, which separates the 5-42 MHz RF input into N separate channels. Each of the N separate channels is provided to a separate QPSK/16-QAM burst receiver channel 58.

Each separate QPSK/16-QAM burst receiver channel 58 is in electrical communication with the headend MAC 60. The headend MAC 60 is in electrical communication with backplane interface 62 which provides an interface to ROM 70, RAM 68, CPU 66, and 100BASE-T Ethernet interface 64. The headend MAC 60 provides clock and a data output to the downstream modulator 72 which

provides an output to amplifier 76 through surface acoustic wave (SAW) filter 74. Amplifier 76 provides an output to 44 MHz IF output, which in turn provides an output to the RF upconverter 78.

Each burst receiver 58 is configured so as to be capable of receiving both QPSK (4-QAM) or 16-QAM signals. The QPSK signals provide 2 bits per symbol, wherein each bit has ±1 amplitude levels. The 16-QAM signals provide 4 bits per symbol, each bit having a ±1 or ±3 amplitude level.

However, the description and illustration of a burst receiver configured to accommodate QPSK and 16-QAM inputs is by way of illustration only and not by way of limitation. Those skilled in the art will appreciate that other modulation techniques, such as 32-QAM, 64-QAM and 256-QAM may alternatively be utilized.

Sample and Packet Synchronization

In addition to the above-mentioned standard request / grant processing, the well-known Data over Cable Service Interface Specifications (DOCSIS)provide for an Unsolicited Grant mode. In accordance with this mode, a fixed number of mini-slots are granted to a selected SID without having to suffer the delay of having a steady stream of requests prior to receipt of corresponding grants. Upstream bandwidth is allocated in discrete blocks at scheduled intervals. The block size and time interval are negotiated between the CM and the CMTS. In other words, given an initial request, the CMTS schedules a steady stream of grants at fixed intervals. The beginning mini-slot of these unsolicited grants will begin a fixed number of mini-slots from the end of the last similar grant. This mechanism can thereby provide a fixed bit rate stream between the CM and CMTS which is particularly useful for packet voice systems which sample the voice at a fixed interval (8kHz) and assemble a fixed

length packet for transport. Such fixed sampling and fixed length packet processing make the use of such fixed grant intervals particularly attractive.

However, if voice samples are collected using an asynchronous clock with respect to the clock associated with the mini-slots, packets will arrive at an arbitrary time with respect to the burst. The time difference (D) between the burst and packet arrival will continuously vary from burst to burst as a function of the difference between the sample and mini-slot clock frequency. FIG. 17 shows the variable delays that result when such voice services are transmitted using the DOCSIS Unsolicited Grant mode. Sample packets (Si, Si+1, ...) arrive based upon the sample clock and upstream grants (G, G+1, ...) arrive based upon the network clock derived from the CMTS network clock. The delay (Di, Di+1, ...) between the sample packet available and the grant arrival varies with every packet as a function of the difference between the sample and network clocks.

However, DOCSIS systems generate a clock used to synchronize the upstream transmission functions. A protocol is defined that provides a synchronized version of the CMTS clock at each CM modem, as has been described in detail hereinabove. A protocol can also be defined that provides synchronization between the voice sample clock and the CM. Similarly, when the Headend communicates with the PSTN through a PSTN Gateway which has its own clocking, a protocol can also be defined that provides synchronization between the PSTN and the Headend. Accordingly, synchronization can then be provided such that the caller voice sampling is synchronized with the CM, which, in turn, is synchronized with the CMTS, which, in turn, is synchronized with the PSTN, ultimately allowing the called destination to be synchronized with the caller. The present invention provides such synchronization.

37110/RJP/B600

Referring to FIG. 18 there is depicted as in FIG. 17, a series of grants (G, G+1,...) and a series of voice samplings (Si, Si+1, ...) wherein the delays (Di, Di+1, ...) between the sample packet arrival and the unsolicited grant arrival is fixed. The fixed delay is a result of synchronization between the CM and the local telephone system as hereinbelow described. The fixed delay is arbitrary and is determined by the random relationship between the start of the call event and the grant timing. It is desirable, however, to minimize the delay between the packet arrival and the grant arrival as set forth in FIG. 19.

In accordance with the present invention, a coordination is provided between the grant arrival processing and the packet arrival assembly processing to help minimize such delay.

The arrival of the grant signal at the CM indicates that "it now is the time for the CM to send the data". Therefore, when the grant arrives the data must be ready for transmission. To prepare data ready for transmission time is needed for both data collection (sampling of the voice) and processing of the collected data (e.g.,providing voice compression). To minimize delay the data for transmission should be ready to transmit just before the grant arrives. Delay occurs if the data collection and processing of the collected data finishes too early and the system has to wait for the grant to arrive. Such a delay can be particularly troublesome for Voice over IP processing which has certain maximum delay specification requirements. Therefore, it is advantageous for the system to know how much time is necessary to collect the data, to know how much time is necessary to process the data, and thereby be able to synchronize such data collection and processing with the grant.

The downstream CM negotiates a grant period with the CMTS as has been hereinabove described. An Unsolicited Grant interval is set by the CMTS, e.g., at 10 ms intervals. Once the grant is established based upon a request (e.g., a signal being sent by

37110/RJP/B600

a caller telephone that a telephone call is desired to be made on an open telephone channel to a call recipient, such as a used connected to the PSTN), the Unsolicited Grant will be provided, namely, the grant will come at regular intervals. The Unsolicited Grant mode is utilized because the voice transmission is continual during the telephone call and is being collected continuously during every grant interval (e.g., every 10 ms). The grant intervals can be considered to be "windows" to transmit the sampled packets of data being collected. However, if the data collection and processing is not synchronized it will not be ready at regular intervals, creating both transmission delay and, in turn, end point (i.e., the call recipient) reception delay.

Referring to FIG. 20, there is depicted a representative embodiment of an implementation of the present invention wherein a local caller can place a call, over a CM / CMTS system, to a call recipient 2002 connected to the PSTN through PSTN gateway 2004. In the representative embodiment, four caller telephones 1047a, 1047b, 1047c, 1047d for part of an analog to digital signal processing system 2010, which is well known to those skilled in the art. Each caller telephone is connected to respective standard code / decode (CODEC) and subscriber loop interface circuits (SLIC), 2012a, 2012b, 2012c, 2012d, which are part of a transmit analog-to-digital (A/D) and receive digital-to-analog (D/A) converter sub-system 2014, which also includes respective buffers 2016a, 2016b, 2016c and 2016d for storing the digital sampled data, and multiplexer/demultiplexer 2018. Converter sub-system 2014 interfaces with a Digital Signal Processor (DSP) 2020, such as LSI Logic Corporation model ZSP16402. DSP 2020 controls signal compression. For example, for transmission, when a caller (e.g., caller 1047a) picks up a telephone receiver and talks, in practice the voice is sampled the converted from analog to digital signals. The DSP controls

37110/RJP/B600

the compression of the data, which is packetized and transmitted under the control of CM 1046 from CM 1046 to CMTS 1042 as hereinabove described. Similarly, for reception, an incoming digital signal gets received and depacketized under the control of CM 1046 and decompressed under the control of the DSP. The resulting digital signals then get converted to analog signals for listening to by the caller.

When a telephone call is to be made through the CM, the telephone being picked up causes a message to be sent to the CMTS requesting an unsolicited grant, e.g., a periodic grant at a 10ms grant period. Voice data is then collected and processed during every 10ms interval between grants. The processing involves the DSP taking the digital signal from the converter sub-system and compressing the digital data (e.g., via an ITU standard G.729 algorithm coder) to enable the use of less bandwidth to transmit. The A/D conversion of a sequence of samples and their buffer storage can be considered the "data collection" aspect. The processing of the collected data has a time established by the compression algorithm chosen. Table 1 below depicts DSP processing time given a 10ms data collection frame size for various ITU compression algorithms using a typical DSP e.g., LSI Logic Corporation model ZSP16402 140 MHz DSP.

Compression	DSP Processing Time
(a) G.711	2 MIPS = 1.4% DSP load = 0.0282 ms to process 2.0 ms of data
(b) G.722	16 MIPS = 11.4% DSP load = 0.228 ms to process 2.0 ms of data
(c) G.726	16 MIPS = 11.4% DSP load = 0.228 ms to process 2.0 ms of data

37110/RJP/B600

(d) G.728	35 MIPS = 25% DSP load = 0.5 ms to process 2.0 ms of data
(e) G.729	20 MIPS = 14.29% DSP load = 1.1 ms to process 2.0 ms of data

TABLE 1

Therefore, for G.711 compression, for example, 2.0ms of data collection time plus 0.0282ms of data processing time, i.e., 2.0282ms, is needed to make the collected and processed data ready just prior to the grant arrival. As such, the data collection must be started at 2.0282ms before the grant arrives and data collection must be finished prior to 0.0282 ms before the grant arrives. In other words, given the grant arrival schedule and the DSP processing time required based upon the compression chosen, clock synchronization between the grant arrival schedule and the data collection deadline is established. To ensure that the data collection deadline is met a clock for the A/D conversion is derived based upon the clocks of the CMTS and CM system and a pointer is provided to indicate a cutoff portion of the buffer in which the sampled data is being collected.

In accordance with the present invention data to be collected (sampling) is based upon the CMTS clock sent from the CMTS synchronizing the CMs. Grant time calculation circuitry 2022 interfaces between DSP 2020 and CM 1046. Collected data is taken from the respective buffer to include data stored in the buffer which was accumulated for a period before grant arrival, namely the processing time plus the data collection time. The CODEC/SLIC has clock to collect the data. The voice sampling is thereby clocked based upon a sample clock signal from the CM.

37110/RJP/B600

As such, the most recent data stored in the buffer just before the grant arrival is used for transmission pursuant to the grant. The details of the sample clocking are set forth below.

Briefly referring back to FIG. 20, call recipient 2002 is connected to the PSTN over well-known PSTN telephone gateway 2004. PSTN telephone gateway 2004 is clocked by a telephony network clock signal 2006 from network clock reference 2003 which is also coupled to CMTS 1042 such that PSTN telephone gateway 2004 can be synchronized with the CMTS clock for the transfer of telephone sample packets 2007 between CMTS 1042 and PSTN telephone gateway 2004. The telephony network clock is the well known Building Integrated Timing Supply (BITS) clock. The equipment requirements for interfacing to this clock are known to those skilled in art and are described in Bellcore document TR-NWT-001244. The concept for intraoffice synchronization is also known to those skilled in the art and is described in Bellcore document TA-NWT-000436. The CMTS clock is synchronized with the telephony network clock signal 2006 via headend clock synchronization which utilizes headend reference tick clock 15, as described above with respect to FIG. 11.

Referring now to FIG. 21, the operation of headend clock synchronization circuit 18 is further described in conjunction with the telephony network clock. Digital tracking loop 2021 is a substantially stable clock output for the CMTS 1042. A difference between an absolute time reference and the output of a local time reference 2022, which is derived from the numerically controlled oscillator 2024, is formed by differencing circuit 2026. This difference defines a frequency error value which represents the difference between the clock of the CMTS 1042 (which is provided by local time reference 2022) and the clock of the PSTN Telephone Gateway 2004 (which is provided by telephony network clock signal 2006). This frequency error value is filtered by loop averaging filter 2028 which prevents

37110/RJP/B600

undesirable deviations in the frequency error value from affecting the numerically controlled oscillator 2024 in a manner which would decrease the stability thereof or cause the numerically controlled oscillator 2024 to operate at other than the desired frequency. The loop filter 2028 is configured so as to facilitate the rapid acquisition of the frequency error value, despite the frequency error value being large, and then to reject comparatively large frequency error values as the digital tracking loop 2021 converges, i.e., as the output of the local timing reference 2022 becomes nearly equal to the absolute time reference, thereby causing the frequency error value to approach zero. Timing offset correction 2030 is a simple adder coupled to local time reference 2022 to time based message generator 2032 which provides time based messages as output. The CMTS clock is now synchronized with the PSTN Gateway clock.

Referring again briefly back to FIG. 20, it is noted that grant time calculation circuitry 2023 and CODEC + SLICs 2012a, 2012b, 2012c, 2012d are responsive to a sample clock signal from CM clock synchronization circuitry 2034 of CM 1046. Such sample clock signal provides the clocking synchronization for the voice sampling at 8KHZ derived from 4.096 MHz CM clock (which is synchronized with the CMTS clock, which is, in turn, synchronized with the PSTN clock.

Referring now to FIG. 22, the operation of CM clock synchronization circuit 2034 is described. The operation of CM clock synchronization circuit 2034 is similar to that of headend clock synchronization circuitry 2008. Time stamp detector 2050 detects downstream data including the timebase messages generated by timebase message generator 2032 of the CMTS 1042. Timebase message detector 2050 provides an absolute time reference which is representative of the frequency of the crystal oscillator timing reference 16 of the CMTS 1042. Digital tracking loop 2036 is included to provide a substantially stable clock output. A

1 37110/RJP/B600

difference between an absolute time reference and the output of a local time reference 2038, which is derived from the numerically controlled oscillator 2040, is formed by differencing circuit 2042. This difference defines a frequency error value. This frequency error value is filtered by loop averaging filter 2044 which prevents undesirable deviations in the frequency error value from affecting the numerically controlled oscillator 2040 in a manner which would decrease the stability thereof or cause the numerically controlled oscillator 2040 to operate at other than the desired frequency. The loop filter 2044 is configured so as to facilitate the rapid acquisition of the frequency error value, despite the frequency error value being large, and then to reject comparatively large frequency error values as the digital tracking loop 2036 converges, i.e., as the output of the local timing reference 2038 becomes nearly equal to the absolute time reference, thereby causing the frequency error value to approach zero. Timing offset correction 2052 is a simple adder coupled to local time reference 2038 to feed sample clock generator 2054 which provides a 4.096 MHZ SAMPLE CLOCK for use by grant time calculation circuitry 2023 and CODEC + SLICs 2012a, 2012b, 2012c, 2012d.

Referring now to FIGS. 23a, 23b and 23c there is respectively depicted the 4.096 MHz sample clock generated, a GrantRcv[4] (i.e., a grant present indication) and a GrantRcv[3:0] SID (i.e., a channel number on which the grant is present.

Referring now to FIGS. 24a, 24b, and 24c there is respectively depicted the derived 8 KHz sample clock for voice sampling, the grant Rcv [4] (in a scaled down depiction) and the sampled data.

Referring to FIGS. 25, 26 and 27, grant time calculation circuitry 2023 is shown in more detail. Epoch counter 2060 is pulsed by an 8KHz pulse generated by pulse generator 2062 derived

37110/RJP/B600

from the 4.096 MHz sample clock produced by CM clock synchronization circuitry 2034 in CM 1046.Grant timing queue 2064 is responsive to the 4 bit SID channel number and grant present signal as shown in FIGS. 23a, 23b and 23c. The grant time calculation circuitry interfaces to DSP 2020 and counts between successive Unsolicited Grants. The epoch counter is a 12 bit counter and is advanced by the 4.096 MHz sample clock with 8 khz enable pulse. The grant arrival timing queue is latched by the grant present signal from the CM 1046. This signal is present whenever a grant of interest is present on the upstream. The grant timing queue accepts a 16 bit input, 4 bit of which are the hardware queue number associated with the grant present signal and 12 bit are the Epoch counter value. The DSP can read the current epoch counter value. The result of grant time calculation by grant time calculation circuitry 2023 is the production of a historical map of when grants arrive with respect to the epoch counter value as shown in FIG. 26. Referring more particularly to FIG. 27, grant timing queue 2064 includes logic block SID_REG, SID_SYNC and SID_FILT for capturing SID information. A 16x16 FIFO stores the tick count for each respective grant and its corresponding SID. Each entry in the FIFO contains the SID and gnt_tick_cnt corresponding to the grant arrival. This information allows DSP software to build a table of SIDs and gnt_tick_cnts which allows calculation of an average inter-arrival time for each grant. This information allows the software to then schedule the data processing as shown and described in more detail below with respect to FIG. 29a to ensure having packets ready in time for the grants.

Referring to FIG. 28, the inter-relationship between grant time calculation circuitry 2023, DSP 2020 and buffers 2016a, ... 2016d are shown in more detail. As indicated above, grant time calculation circuitry 2023 provides DSP Data Read Access information (SID and gnt_tick_cnts) to DSP 2020. This DSP Data

37110/RJP/B600

Read Access information provides the timing information to the DSP so that it will know when and where to read the upstream data from the upstream data buffer. It also provides timing information as to when to place the downstream uncompressed voice data into the down stream data buffer. This timing information allows software 2070 for DSP 2020 to build a table 2072 of SIDs and grant tick counts, calculate an average inter-arrival time for each grant, schedules the data processing, and controls data transfers into and out of the data buffers.

As seen in FIG. 28, representative buffer 2016a (e.g., SID/Channel 1) and buffer 2016d (e.g., SID / channel 4) include both an upstream data buffer and a downstream data buffer, each having its respective CODEC/SLIC and clocked by the Sample Clock as described hereinabove. When sampled voice packet data is to be sent along Channel 1, in response to a grant, a Channel 1 data pointer under the control of DSP 2020 utilizes the grant time calculation information from grant time calculation circuitry 2023 to identify from where in the upstream data buffer the most current sampled data is to be taken and transmitted to CM 1046, the not-as-current samples beyond the pointer (i.e., stored earlier in the buffer for Channel 1) is discarded. Similarly, when sampled voice packet data is to be sent along Channel 4, in response to a grant, a Channel 4 data pointer under the control of DSP 2020 utilizes the grant time calculation information from grant time calculation circuitry 2023 to identify from where in the upstream data buffer the most current sampled data is to be taken and transmitted to CM 1046, the not-as-current samples beyond the pointer (i.e., stored earlier in the buffer for Channel 4) is discarded. The selected sampled voice packet data is then transmitted to CM 1046 by DSP 2020 for transmission to CMTS 1042 as hereinabove described.

37110/RJP/B600

Referring to FIGS. 29a and 29b an operational flow chart is provided showing DSP system software decision implementation in accordance with the present invention.

Consider a system where DSP 2020 is a 140 MIPS digital signal processor, such as LSI Logic Corporation model ZSP16402, the transport package (TP) package size is 10ms, i.e., the voice package size in milliseconds within each grant interval that is being transmitted to/from the telephone, and the data processing involves voice compression selected from Table 1 set forth above where the data processing time needed before grant is 2ms for those compression algorithms other than G.729 wherein the time needed is 10ms. In other words, referring back to Table 1, for each 2.0ms, the DSP must encode and decode 4 channels of data while the 10 ms is used for the signaling of a TP package transmission. The far-end voice and the near end voice are synchronized via the sample clock. It should be noted, for example, that it would take 100% of the DSP load to process 4 channels of G.728 for the 140 MIPS DSP.

Referring back to FIG. 29a, at stage 2080, inputs as to Channel Number initiating a request, corresponding grant present and sample clock from cable modem 10 are provided for grant time calculation 2082 and channel assessment start 2084 by the DSP software. A particular channel open, i.e., channel i = 1, 2, 3, or 4, is determined at stage 2086. If no, the processing begins again, if yes, processing time Ti, as seen in FIG. 29b, is set at stage 2088 based upon the compression algorithm chosen. At stage 2090, upon the grant time calculation receipt by the DSP, 2ms of data from the pointer location in the corresponding buffer associated with the open channel is read. For those algorithms with 2ms processing time, five processing cycles, having a j index going from 1 to 5, is needed. For the G.729 algorithm a 2ms processing time cannot be used since the uncompressed voice data is only available at 10ms frame-size. As such, at stage

37110/RJP/B600

2092 a determination as to G.729 is made, and if the determination is no 2ms of data is processed at stage 2094. If there is G.729 compression, the cycle index j is determined at stage 2096, and if, no more data is read incrementally j= j+1 at stage 2098. Once j=5 at stage 2096, 10ms of data is processed at stage 3000 and the 10 ms package is then transmitted at stage 3002 pursuant to the current grant arrival. Similarly to the j indexing for data read, a j indexing is performed for data processing at stages 3004 and 3006. Once the processing index j=5 at stage 3004, where the 5 2ms iterations have been completed, the 10 ms package is sent at stage 3002.

Those skilled in the art will appreciate that alternative embodiments to that which has been described herein can be implemented. For example, while the present invention has been described in conjunction with a cable modem / cable modem termination system, the present invention can be used with any transmission system that allocates bandwidth periodically instead of on demand, such as with the well known Asynchronous Transfer Mode (ATM) protocol system. Further, interrupts could be generated by the hardware to indicate that upstream transmission is complete. This signal would identify the time when the upstream transmission means has sent all of the data and the transmission buffer is now available for re-use.

Those skilled in the art can also appreciate that a method for communicating information, as set forth in more detail in Appendix A hereto, can include:allocating a time slot in a time division multiple access system for a transmission from a subscriber to a headend the time slot being sufficient for only a first portion of a transmission, a second portion of the transmission being transmitted in other than the first time slot; enhancing synchronization a clock of the subscriber with respect to a clock of the headend using a message transmitted from the headend to the subscriber which is indicative of an error in a

1 37110/RJP/B600

subscriber transmission time with respect to the time slot; using a feedback loop process to determine at least one of fractional symbol timing correction and carrier phase correction of a transmission from the subscriber to the headend; monitoring quality of at least one channel and changing modulation in response changes to monitored channel quality; using information representative of parameters of received time division multiple access data to facilitate processing of the received time division multiple access data in a receiver, the parameters being communicated within the headend; and generating filter coefficients at the headend from a ranging signal transmitted from the subscriber to the headend and transmitting the filter coefficients from the headend to the subscriber, the filter coefficients being used of the subscriber to compensate for noise in a transmission from the subscriber to the headend.

Those skilled in the art can also appreciate that such information communication methodology as set forth in the Appendix enclosed herewith, or portions thereof, can be combined with the further methodology described hereinabove with regard to the processing of sampled packets, namely, the processing of sampled packets from a packet sender for transmission over a transmission system having a periodically allocated bandwidth to a packet recipient by: determining unsolicited grant arrivals in response to a request from the packet sender; synchronizing the storing of sampled packets with the unsolicited grant arrivals; and transmitting, upon receipt of an unsolicited grant arrival, currently stored sampled packets for further transmission to the packet recipient over the transmission system having a periodically allocated bandwidth. Such a combined system can provided an enhanced information communication methodology.

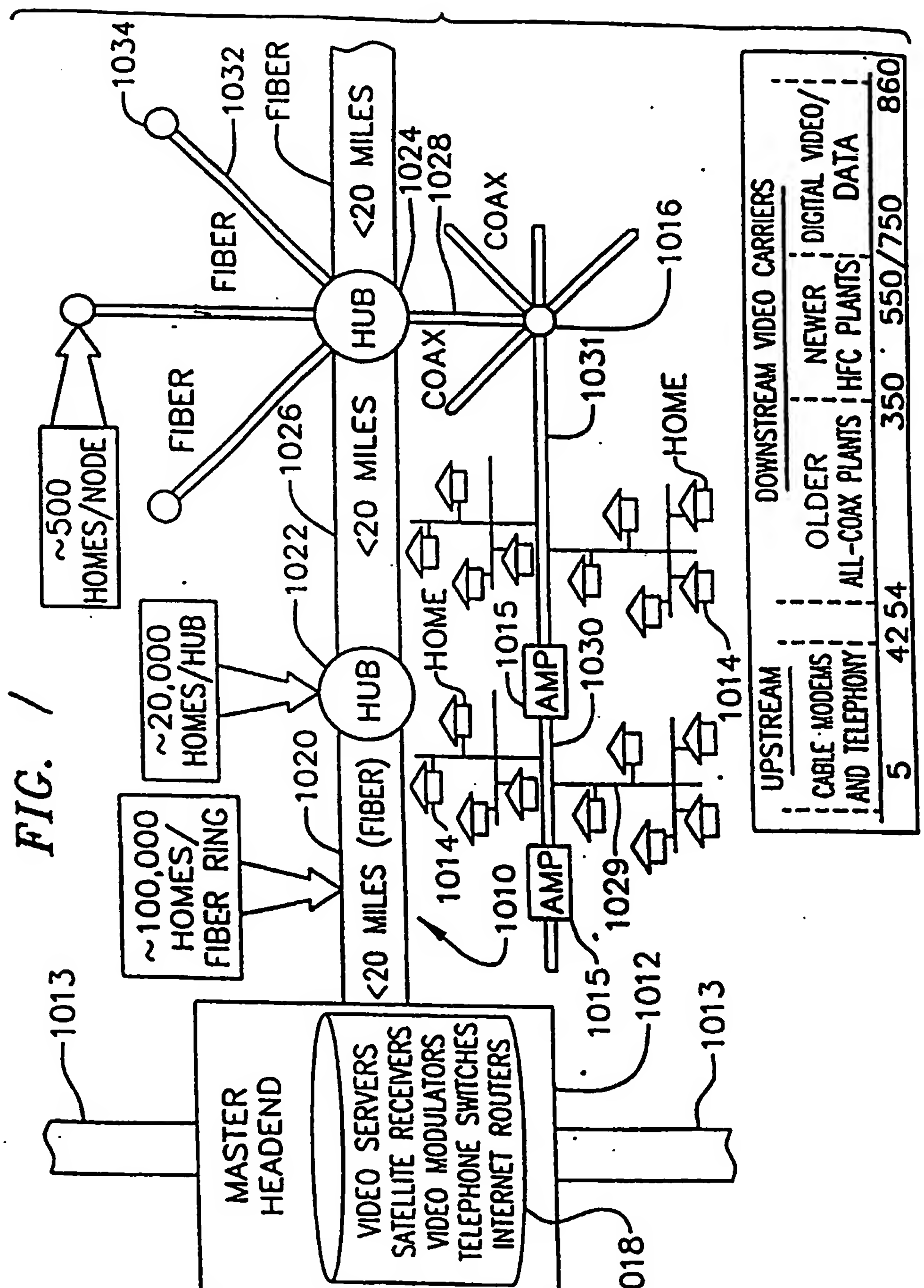

MASTER HEADEND
VIDEO SERVERS
SATELLITE RECEIVERS
VIDEO MODULATORS
TELEPHONE SWITCHES
INTERNET ROUTERS
1018
1013
1013
1012
1015
1015
<20 MILES (FIBER)
~100,000 HOMES/ FIBER RING
1020
~20,000 HOMES/HUB
HUB
1022
<20 MILES
~500 HOMES/NODE
FIBER
FIBER
FIBER
FIBER
1026
1034
1032
1024
1028
<20 MILES
HUB
COAX
COAX
1016
1031
HOME
HOME
HOME
AMP
AMP
1030
1014
1014
1029
1010
UPSTREAM
CABLE MODEMS AND TELEPHONY
DOWNSTREAM VIDEO CARRIERS
OLDER ALL-COAX PLANTS
NEWER HFC PLANTS
DIGITAL VIDEO/ DATA
5
42
54
350
550/750
860

FIG. 1

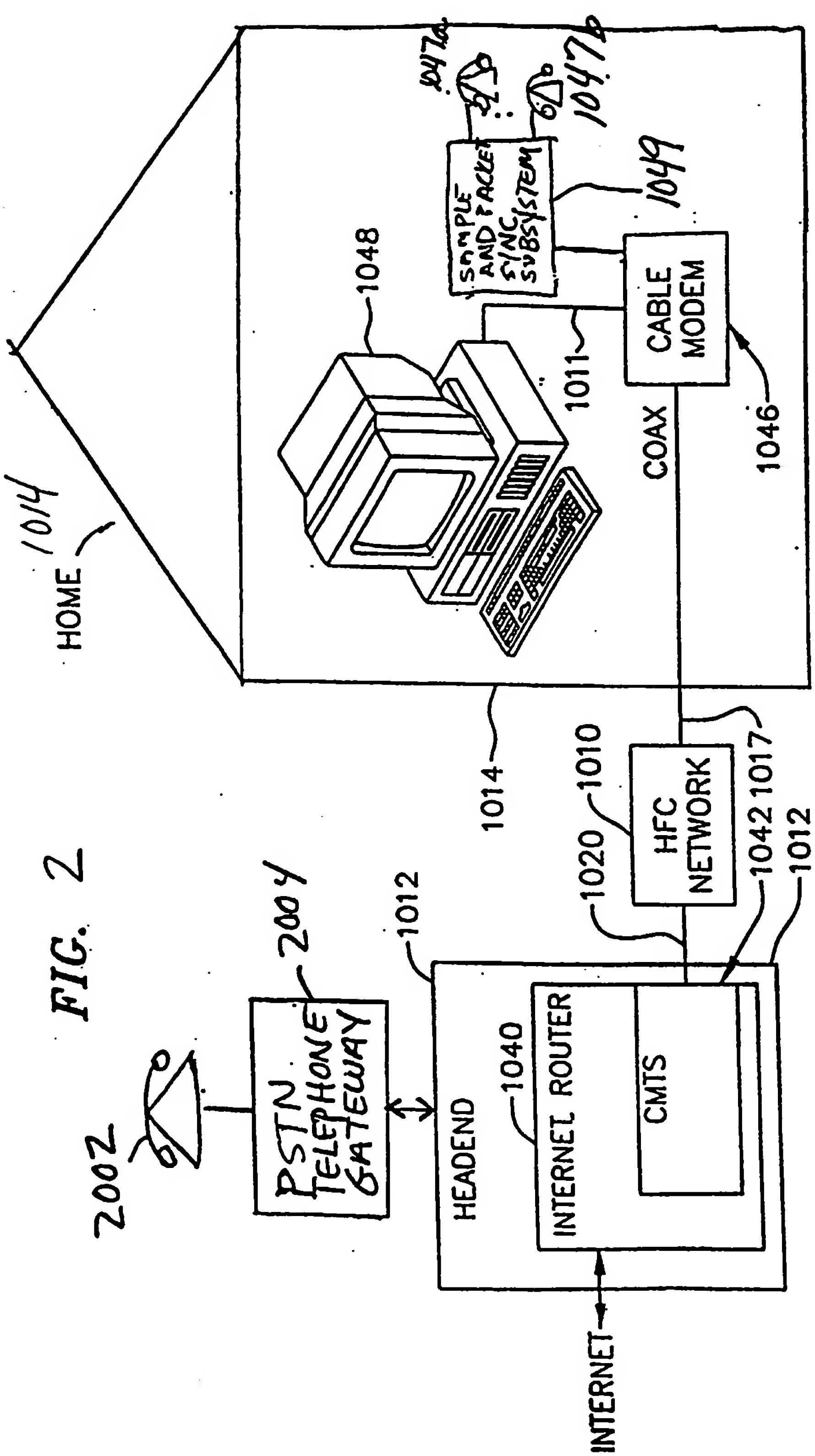

HOME
1014
1048
SAMPLE AND PACKET SYNC SUBSYSTEM
1047a
1047b
1049
1011
CABLE MODEM
COAX
1046
1014
1010
HFC NETWORK
1020
1042
1017
1012
1012
HEADEND
1040
INTERNET ROUTER
CMTS
PSTN TELEPHONE GATEWAY
2004
2002
INTERNET

FIG. 2

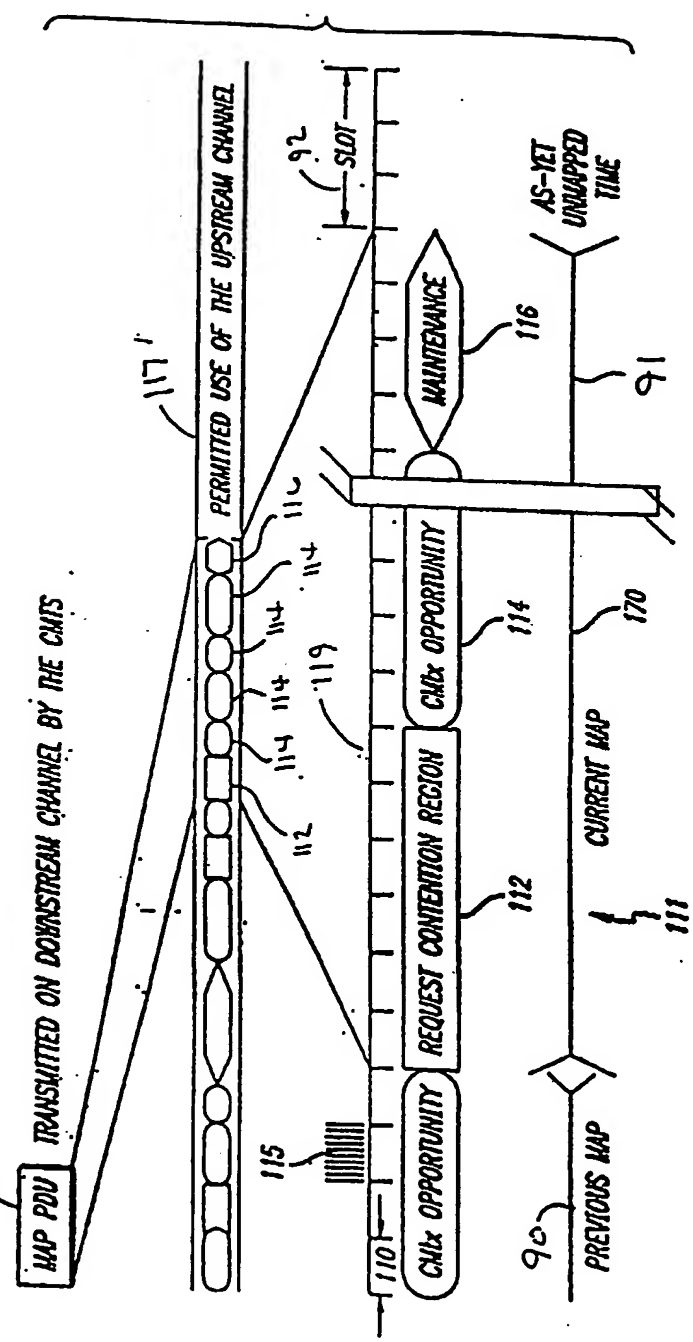

113
MAP PDU
TRANSMITTED ON DOWNSTREAM CHANNEL BY THE CMTS
117
PERMITTED USE OF THE UPSTREAM CHANNEL
92
SLOT
115
110
CMx OPPORTUNITY
REQUEST CONTENTION REGION
112
119
CMx OPPORTUNITY
114
MAINTENANCE
116
90
PREVIOUS MAP
111
CURRENT MAP
170
91
AS-YET UNMAPPED TIME

FIG. 3

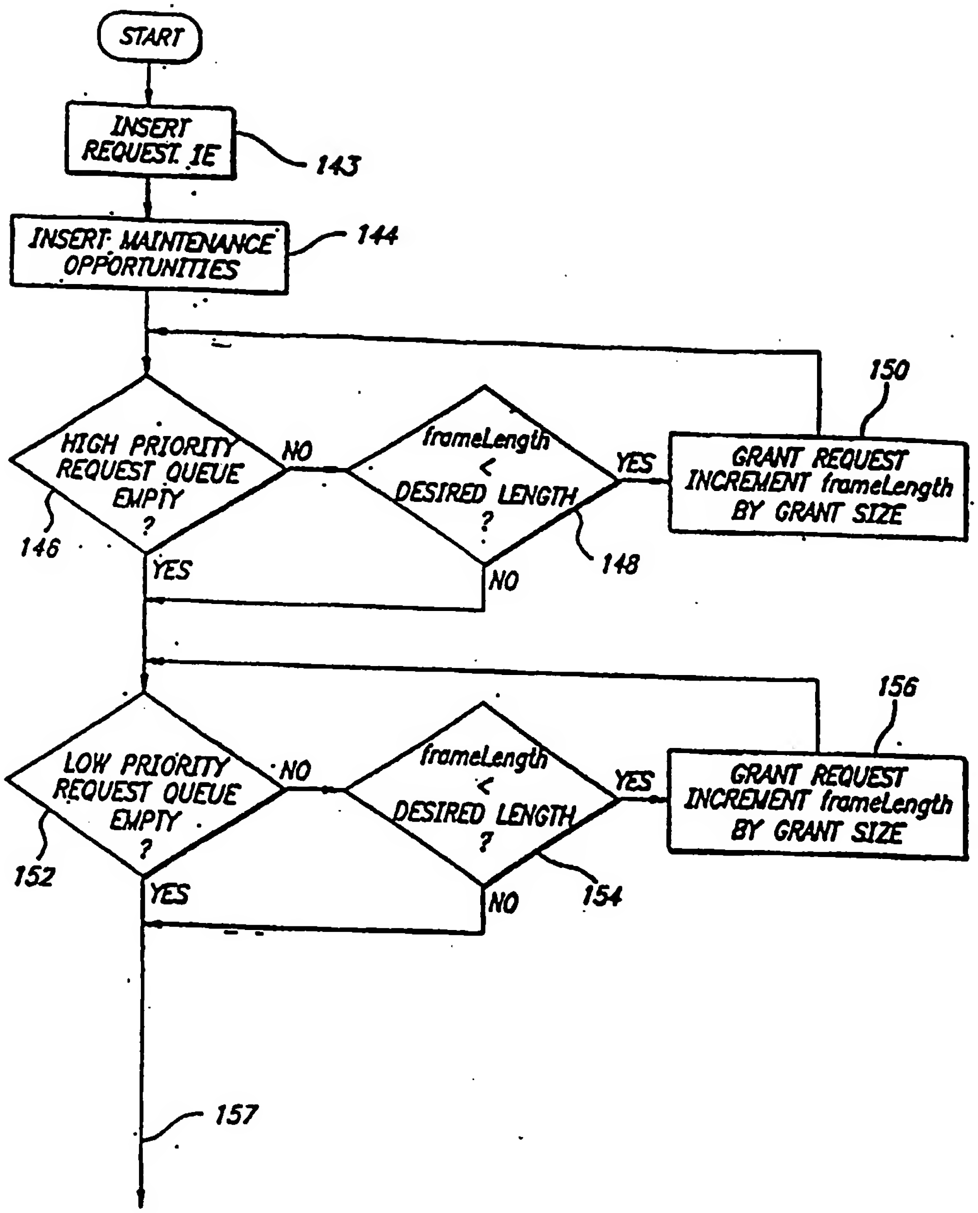

START
INSERT REQUEST IE
143
INSERT MAINTENANCE OPPORTUNITIES
144
HIGH PRIORITY REQUEST QUEUE EMPTY ?
146
NO
frameLength < DESIRED LENGTH ?
148
YES
GRANT REQUEST INCREMENT frameLength BY GRANT SIZE
150
YES
NO
LOW PRIORITY REQUEST QUEUE EMPTY ?
152
NO
frameLength < DESIRED LENGTH ?
154
YES
GRANT REQUEST INCREMENT frameLength BY GRANT SIZE
156
YES
NO
157

FIG. 4

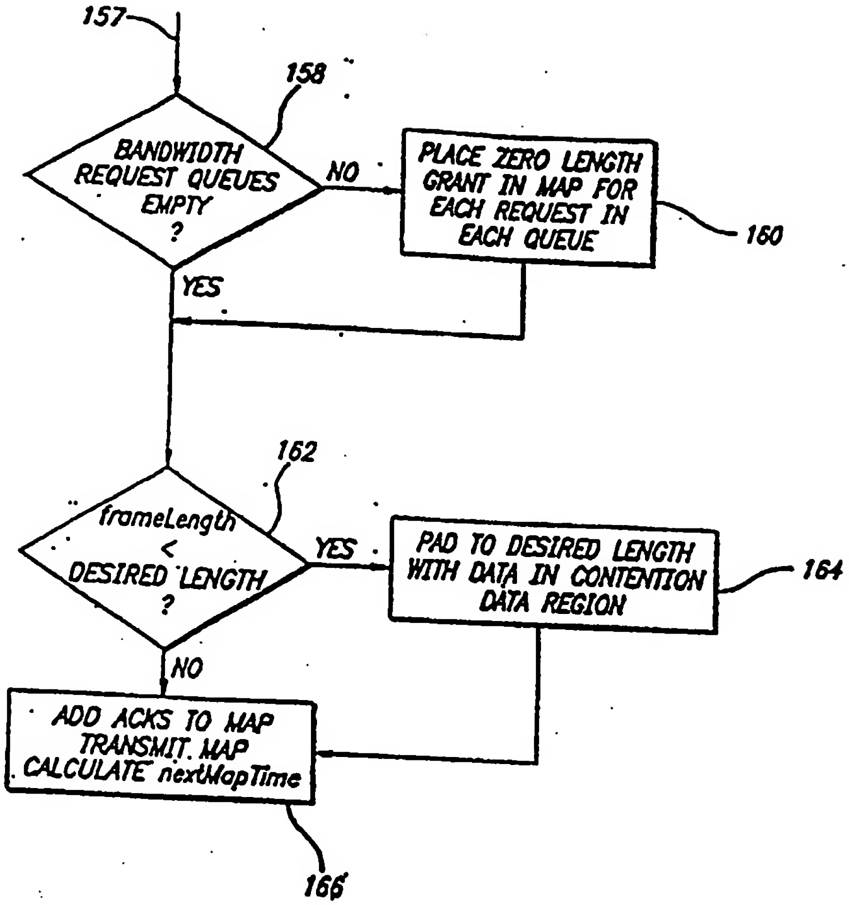

157
158
BANDWIDTH REQUEST QUEUES EMPTY ?
NO
PLACE ZERO LENGTH GRANT IN MAP FOR EACH REQUEST IN EACH QUEUE
160
YES
162
frameLength < DESIRED LENGTH ?
YES
PAD TO DESIRED LENGTH WITH DATA IN CONTENTION DATA REGION
164
NO
ADD ACKS TO MAP TRANSMIT MAP CALCULATE nextMapTime
166

FIG. 5

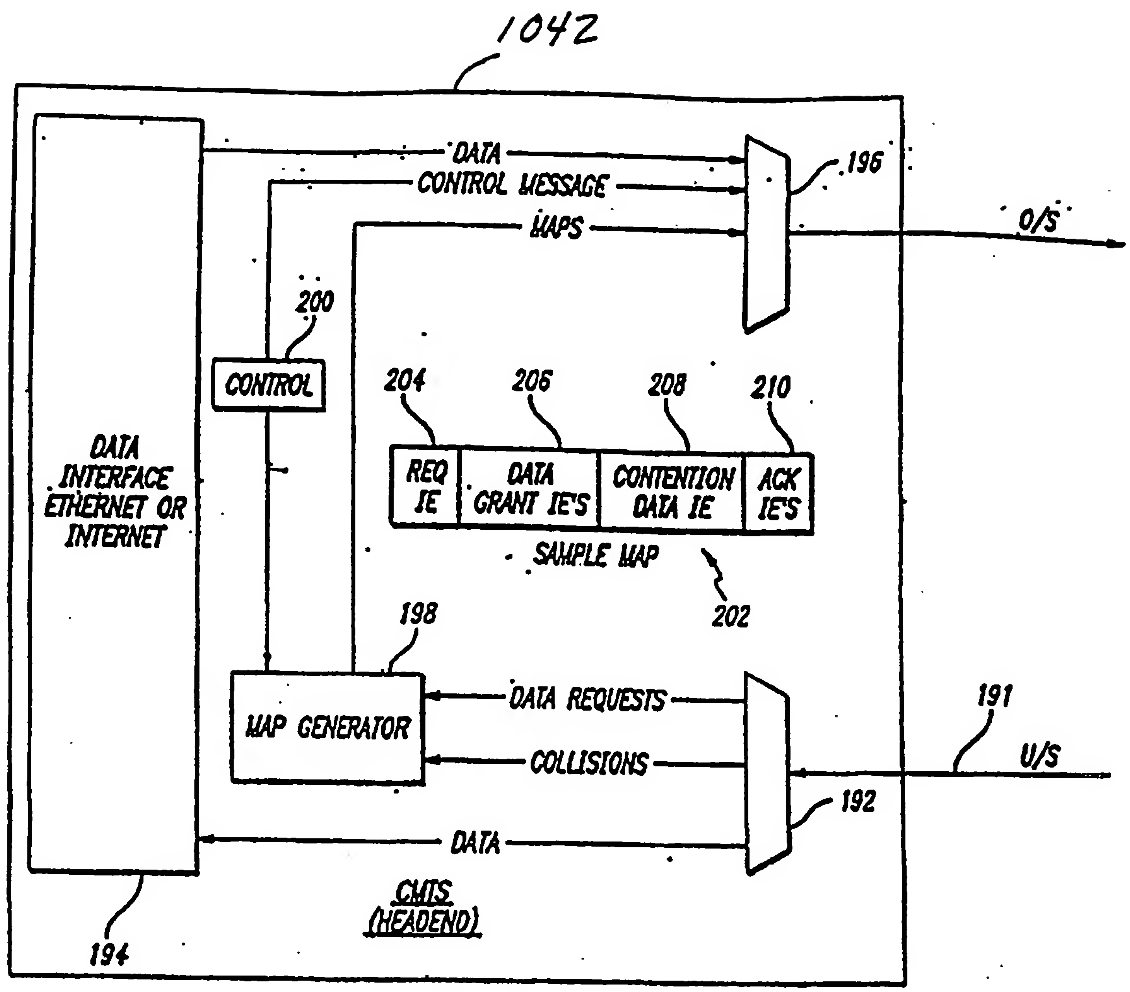

1042
DATA
CONTROL MESSAGE
MAPS
196
O/S
200
CONTROL
204
206
208
210
REQ IE
DATA GRANT IE'S
CONTENTION DATA IE
ACK IE'S
SAMPLE MAP
202
DATA INTERFACE ETHERNET OR INTERNET
198
MAP GENERATOR
DATA REQUESTS
COLLISIONS
191
U/S
192
DATA
CMTS (HEADEND)
194

FIG. 6

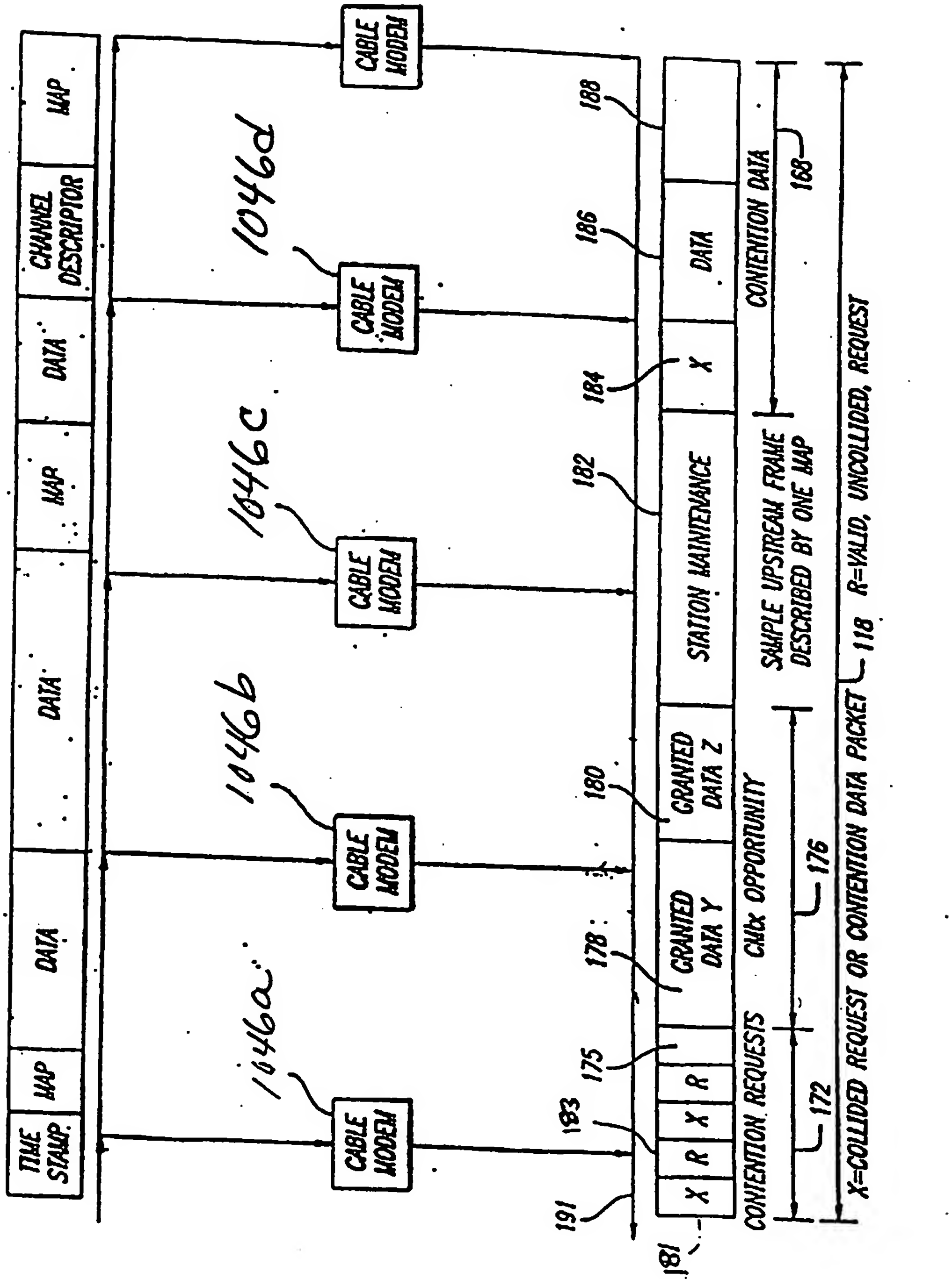

TIME STAMP
MAP
DATA
DATA
MAP
DATA
CHANNEL DESCRIPTOR
MAP
CABLE MODEM
CABLE MODEM
CABLE MODEM
CABLE MODEM
CABLE MODEM
1046a
1046b
1046c
1046d
191
181
183
175
178
180
182
184
186
188
X
R
X
R
GRANTED DATA Y
GRANTED DATA Z
STATION MAINTENANCE
X
DATA
CONTENTION REQUESTS
OPPORTUNITY
CONTENTION DATA
172
176
168
SAMPLE UPSTREAM FRAME DESCRIBED BY ONE MAP
X=COLLIDED REQUEST OR CONTENTION DATA PACKET
118
R=VALID, UNCOLLIDED, REQUEST

FIG. 7

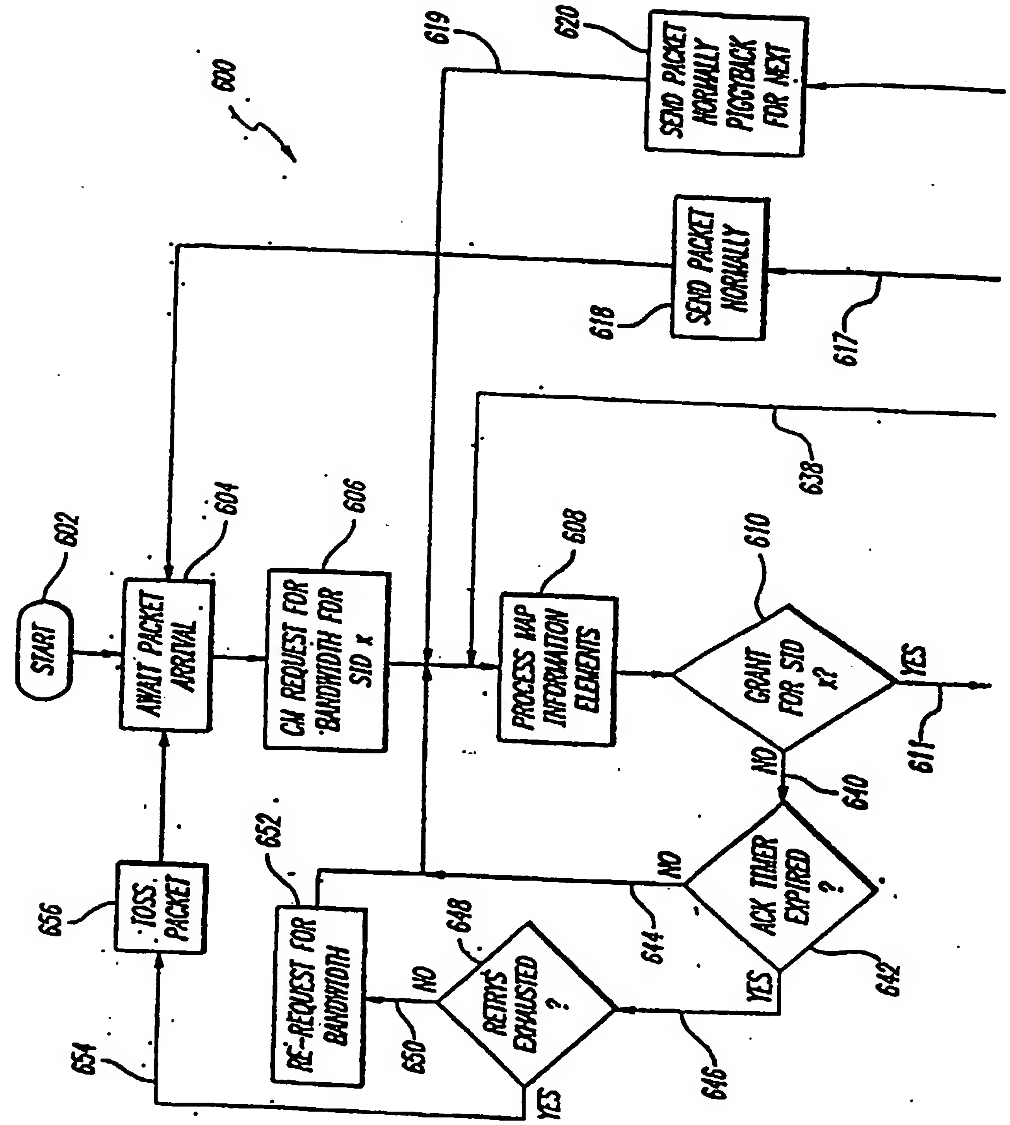

600
START
602
AWAIT PACKET ARRIVAL
604
CM REQUEST FOR BANDWIDTH FOR SID x
606
PROCESS MAP INFORMAITON ELEMENTS
608
GRANT FOR SID x?
610
YES
611
NO
640
ACK TIMER EXPIRED ?
642
NO
644
YES
646
RETRYS EXHAUSTED ?
648
NO
650
YES
RE-REQUEST FOR BANDWIDTH
652
654
TOSS PACKET
656
638
SEND PACKET NORMALLY
618
617
SEND PACKET NORMALLY PIGGYBACK FOR NEXT
620
619

FIG. 8

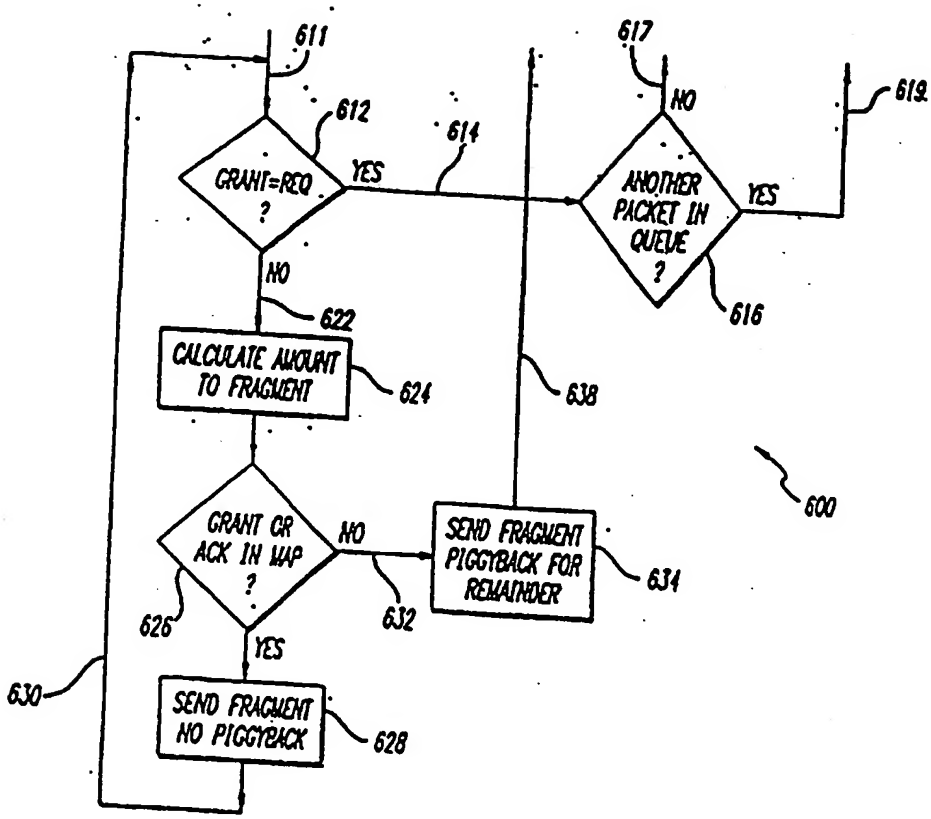
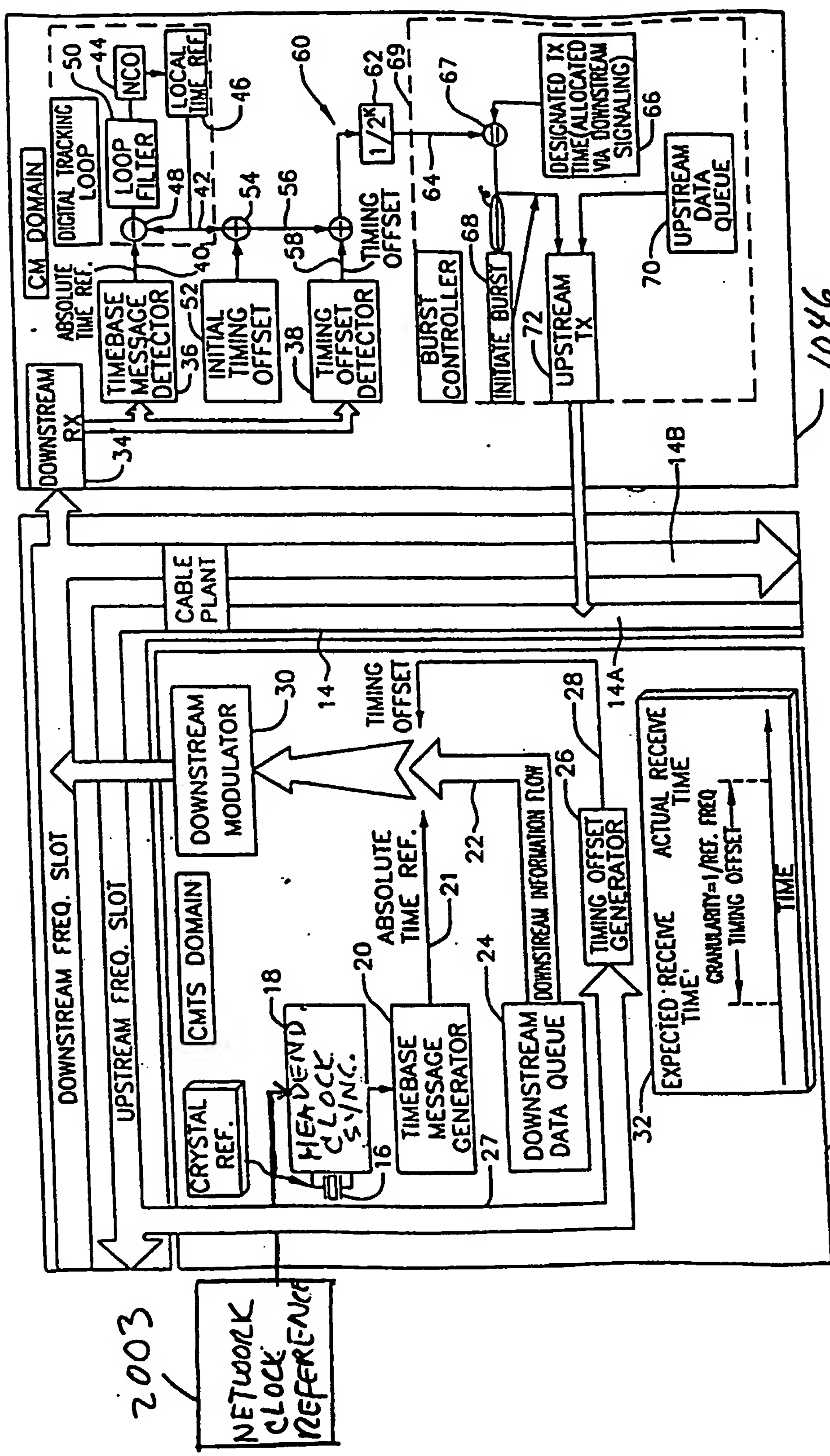

611
612
GRANT=REQ ?
YES
614
617
NO
ANOTHER PACKET IN QUEUE ?
YES
619
616
NO
622
CALCULATE AMOUNT TO FRAGMENT
624
638
GRANT OR ACK IN MAP ?
NO
632
SEND FRAGMENT PIGGYBACK FOR REMAINDER
634
600
626
YES
630
SEND FRAGMENT NO PIGGYBACK
628

FIG. 9

FIG. 1Q


CM DOMAIN
DIGITAL TRACKING LOOP
DOWNSTREAM RX
TIMEBASE MESSAGE DETECTOR
ABSOLUTE TIME REF.
INITIAL TIMING OFFSET
TIMING OFFSET DETECTOR
TIMING OFFSET
LOOP FILTER
NCO
LOCAL TIME REF
1/2K
BURST CONTROLLER
INITIATE BURST
UPSTREAM TX
DESIGNATED TX TIME (ALLOCATED VIA DOWNSTREAM SIGNALING)
UPSTREAM DATA QUEUE
CABLE PLANT
DOWNSTREAM FREQ. SLOT
UPSTREAM FREQ. SLOT
CMTS DOMAIN
CRYSTAL REF.
HEADEND CLOCK SYNC.
TIMEBASE MESSAGE GENERATOR
DOWNSTREAM DATA QUEUE
DOWNSTREAM INFORMATION FLOW
DOWNSTREAM MODULATOR
TIMING OFFSET GENERATOR
EXPECTED RECEIVE TIME
ACTUAL RECEIVE TIME
GRANULARITY=1/REF. FREQ
TIMING OFFSET
TIME
NETWORK CLOCK REFERENCE
2003
1042
1046
14
14A
14B
16
18
20
21
22
24
26
27
28
30
32
34
36
38
40
42
44
46
48
50
52
54
56
58
60
62
64
66
67
68
69
70
72

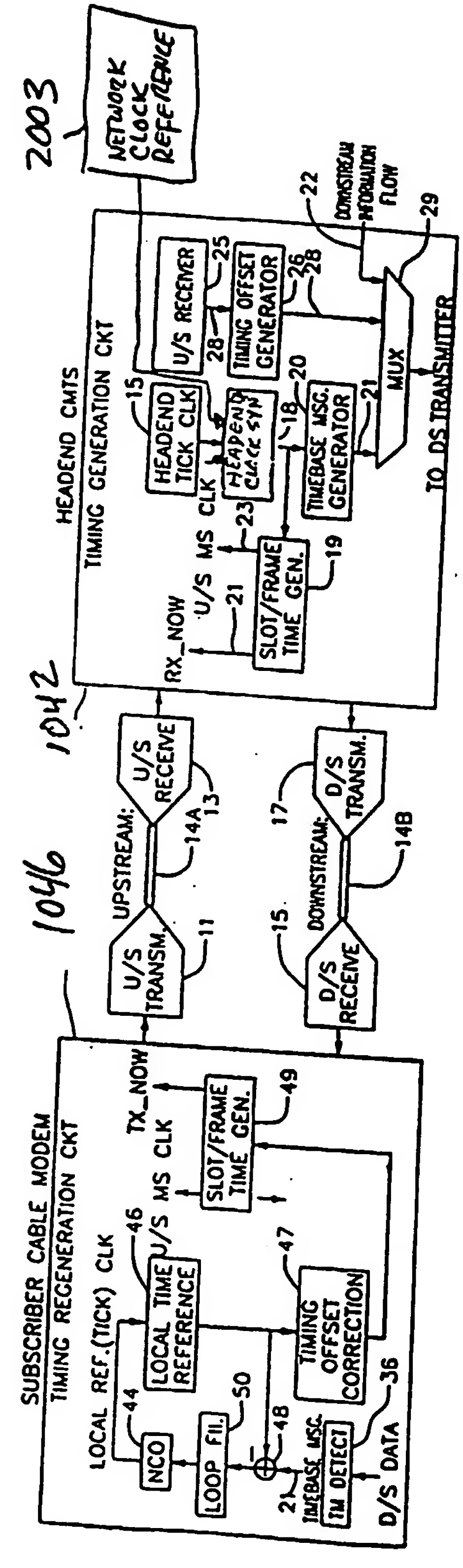

NETWORK CLOCK REFERENCE
2003
HEADEND CMTS
TIMING GENERATION CKT
HEADEND TICK CLK
HEADEND CLOCK SYN
U/S RECEIVER
TIMING OFFSET GENERATOR
TIMEBASE MSG. GENERATOR
SLOT/FRAME TIME GEN.
U/S MS CLK
RX_NOW
MUX
DOWNSTREAM INFORMATION FLOW
TO DS TRANSMITTER
U/S RECEIVE
D/S TRANSM.
UPSTREAM:
DOWNSTREAM:
U/S TRANSM.
D/S RECEIVE
1042
1046
SUBSCRIBER CABLE MODEM
TIMING REGENERATION CKT
LOCAL REF. (TICK) CLK
NCO
LOOP FIL.
LOCAL TIME REFERENCE
TIMING OFFSET CORRECTION
TIMEBASE MSG. TM DETECT
D/S DATA
SLOT/FRAME TIME GEN.
U/S MS CLK
TX_NOW

FIG. 11

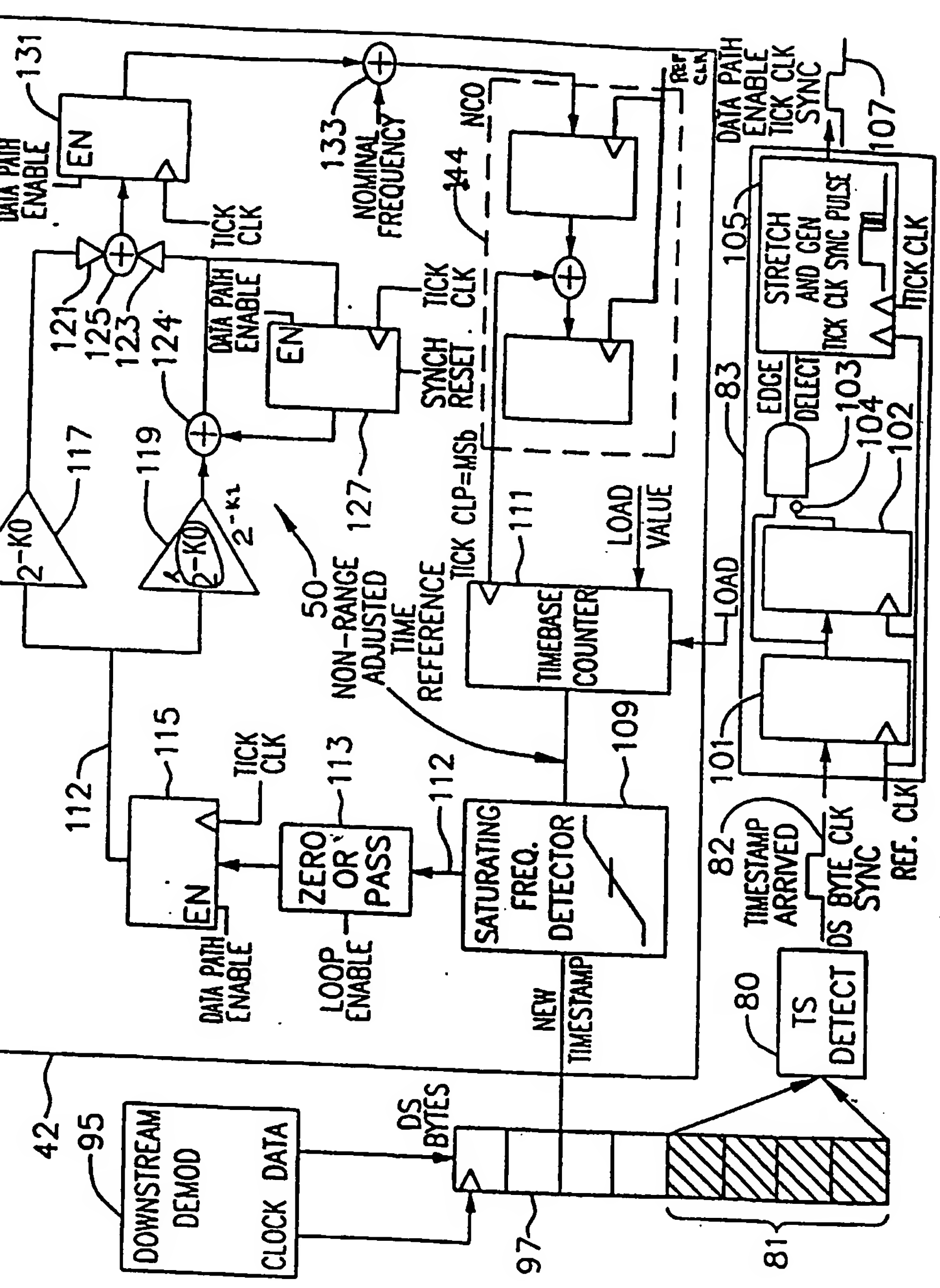

DOWNSTREAM DEMOD
CLOCK DATA
DS BYTES
TS DETECT
TIMESTAMP ARRIVED
DS BYTE CLK SYNC
REF. CLK
NEW TIMESTAMP
SATURATING FREQ. DETECTOR
ZERO OR PASS
LOOP ENABLE
DATA PATH ENABLE
EN
TICK CLK
TIMEBASE COUNTER
LOAD VALUE
LOAD
TICK CLP=MSb
NON-RANGE ADJUSTED TIME REFERENCE
2^{-K0}
2^{-K1}
SYNCH RESET
NCO
NOMINAL FREQUENCY
REF CLK
EDGE DETECT
STRETCH AND GEN TICK CLK SYNC PULSE
DATA PATH ENABLE TICK CLK SYNC
42
95
97
81
80
82
50
101
102
103
104
105
107
83
109
111
112
113
115
117
119
121
123
124
125
127
131
133
144

FIG. 12

FIG. 13

UPDATE RATE	COARSE COEFFS	FINE COEFFICIENTS
1kHz(1ms)	$K0=2^{-11}$ $K1=2^{-15}$ (BW=50Hz)	$K0=2^{-16}$ $K1=2^{-25}$ (BW=1Hz)
300Hz(3.3ms)	$K0=2^{-12}$ $K1=2^{-15}$ (BW=20Hz)	$K0=2^{-16}$ $K1=2^{-23}$ (BW=1Hz)
100Hz(10ms)	$K0=2^{-13}$ $K1=2^{-16}$ (BW=10Hz)	$K0=2^{-16}$ $K1=2^{-22}$ (BW=1Hz)
50Hz(20ms)	$K0=2^{-14}$ $K1=2^{-17}$ (BW=5Hz)	$K0=2^{-16}$ $K1=2^{-21}$ (BW=1Hz)
30Hz(33ms)	$K0=2^{-15}$ $K1=2^{-18}$ (BW=3Hz)	$K0=2^{-17}$ $K1=2^{-21}$ (BW=1Hz)
10Hz(100ms)	$K0=2^{-17}$ $K1=2^{-20}$ (BW=1Hz)	$K0=2^{-17}$ $K1=2^{-20}$ (BW=1Hz)
5Hz(200ms)	$K0=2^{-18}$ $K1=2^{-20}$ (BW=1Hz)	$K0=2^{-18}$ $K1=2^{-20}$ (BW=1Hz)

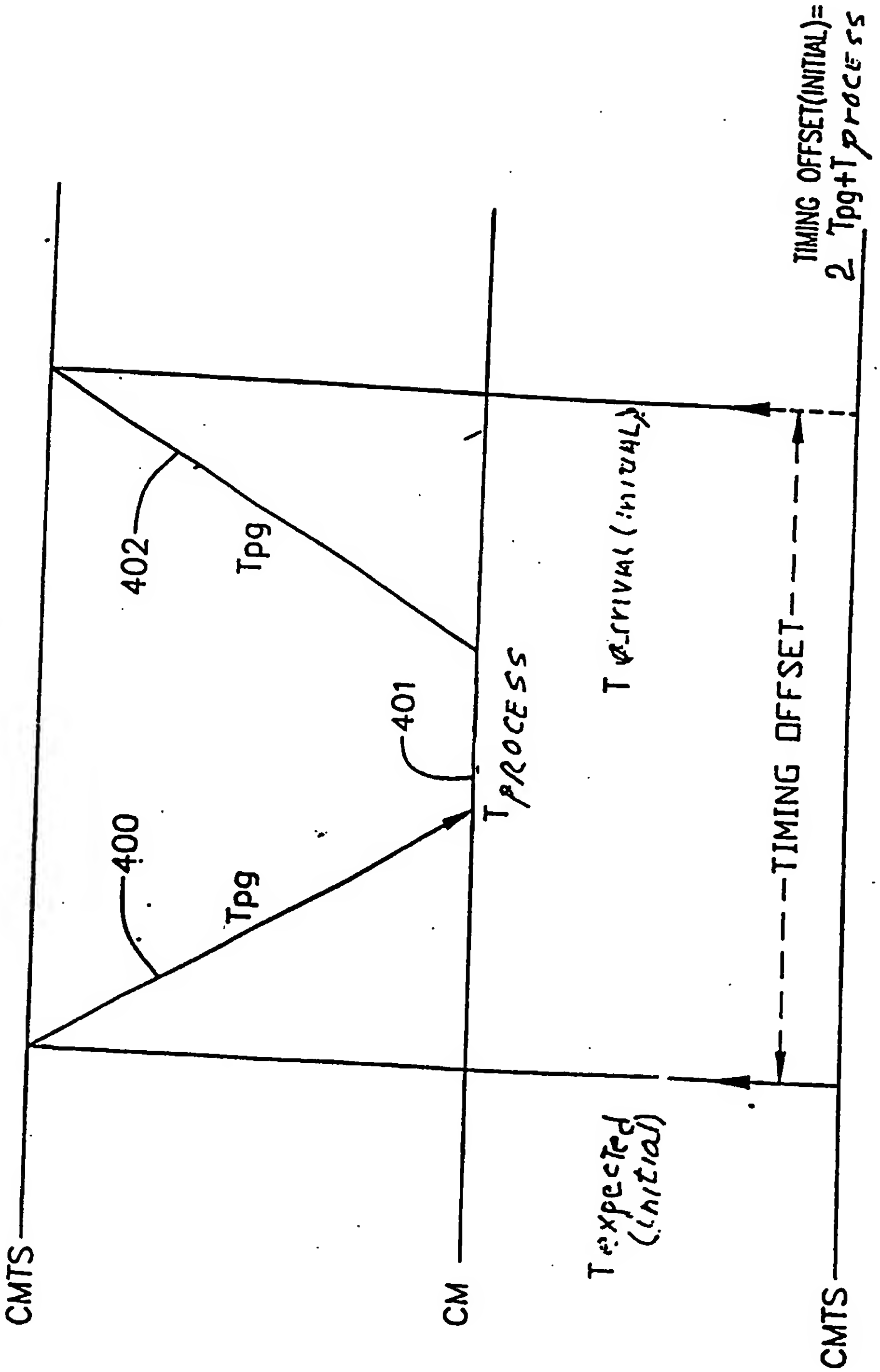

CMTS
CM
CMTS
400
Tpg
401
T PROCESS
402
Tpg
T expected (initial)
T arrival (initial)
TIMING OFFSET
TIMING OFFSET(INITIAL)=
2 Tpg+T process

FIG. 14

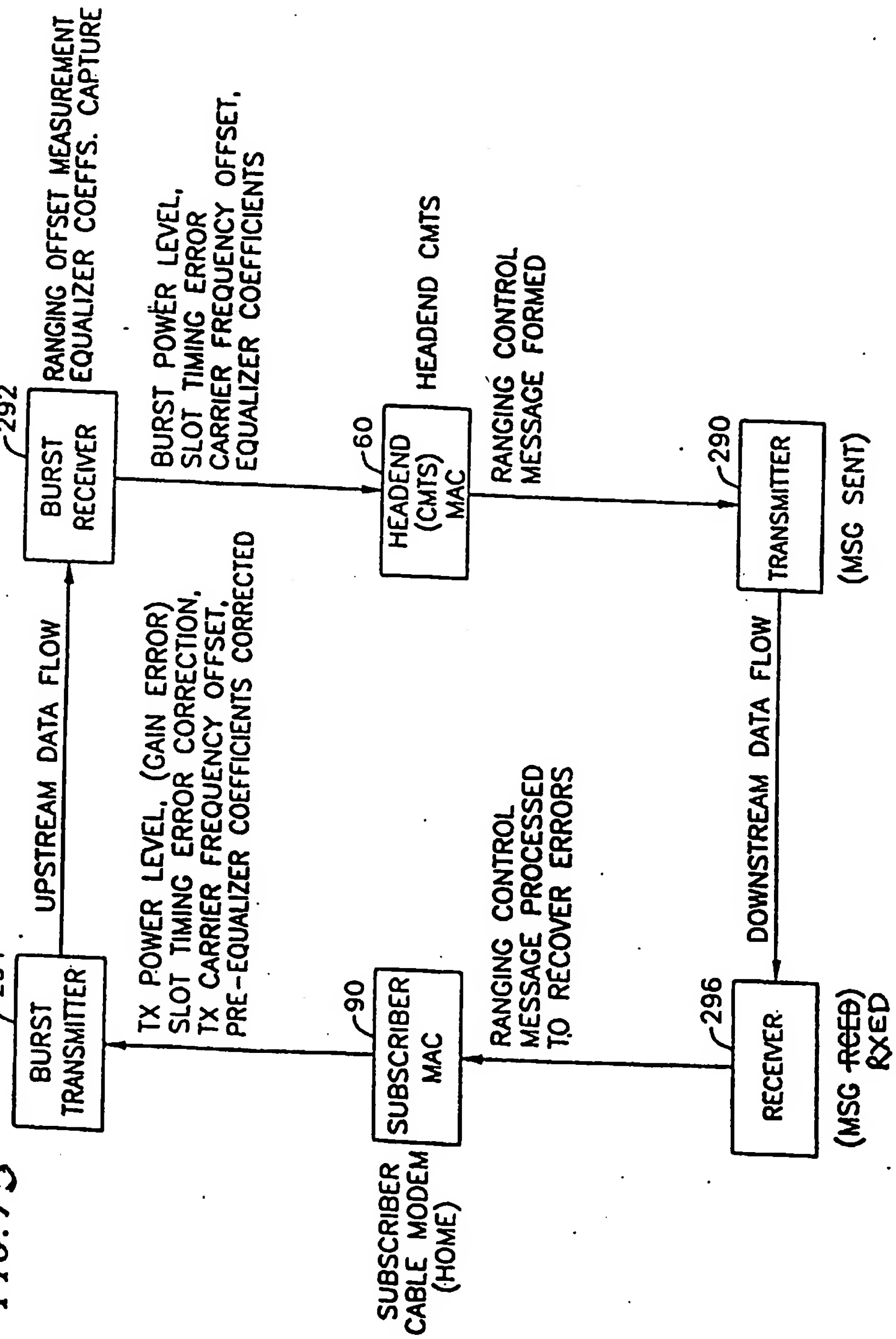

BURST RECEIVER
292
RANGING OFFSET MEASUREMENT
EQUALIZER COEFFS. CAPTURE
BURST POWER LEVEL,
SLOT TIMING ERROR
CARRIER FREQUENCY OFFSET,
EQUALIZER COEFFICIENTS
HEADEND (CMTS) MAC
60
HEADEND CMTS
RANGING CONTROL
MESSAGE FORMED
TRANSMITTER
290
(MSG SENT)
DOWNSTREAM DATA FLOW
RECEIVER
296
(MSG ~~RCED~~ RXED)
RANGING CONTROL
MESSAGE PROCESSED
TO RECOVER ERRORS
SUBSCRIBER MAC
90
SUBSCRIBER
CABLE MODEM
(HOME)
TX POWER LEVEL, (GAIN ERROR)
SLOT TIMING ERROR CORRECTION,
TX CARRIER FREQUENCY OFFSET,
PRE-EQUALIZER COEFFICIENTS CORRECTED
BURST TRANSMITTER
294
UPSTREAM DATA FLOW

FIG. 15

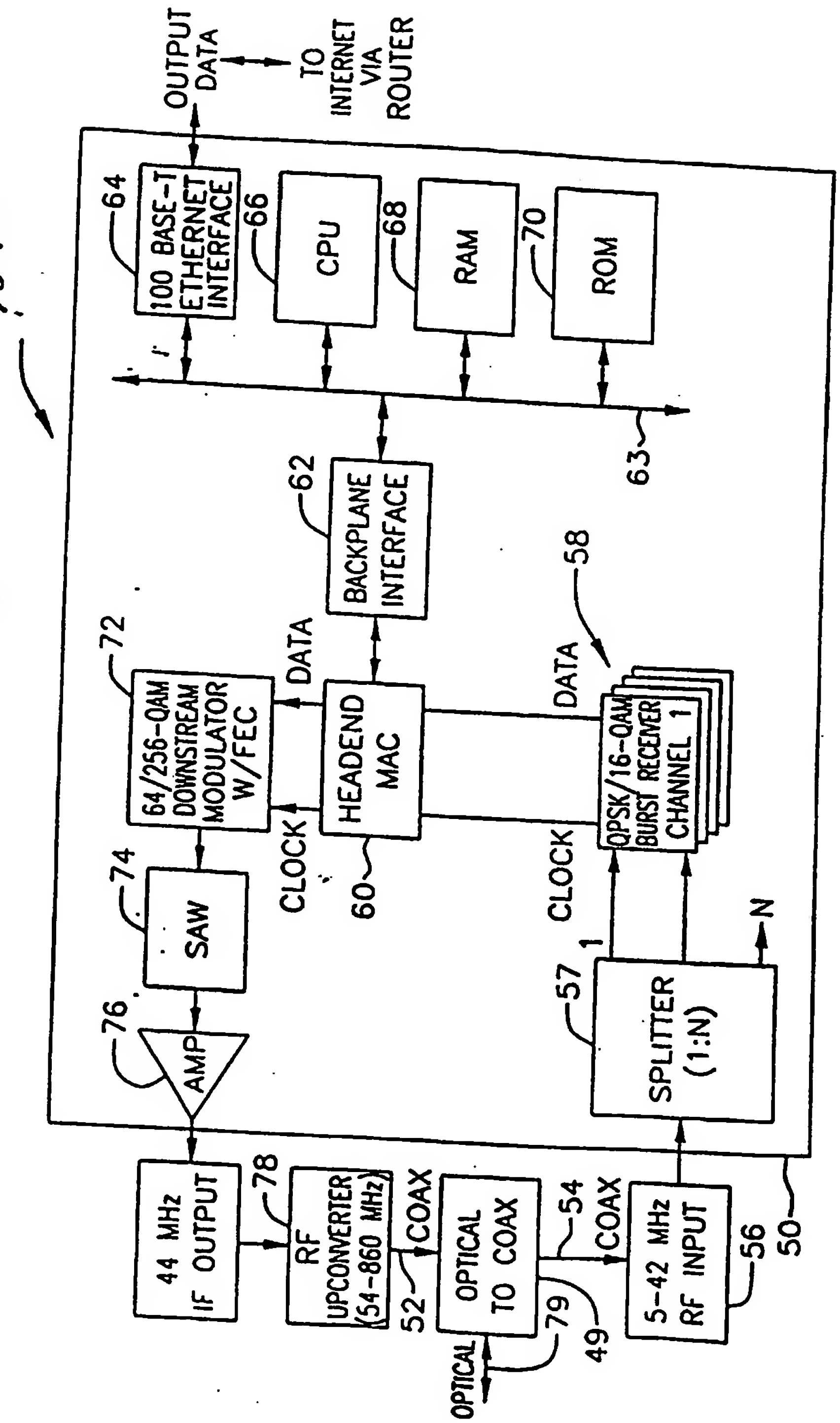

1042
OUTPUT DATA
TO INTERNET VIA ROUTER
100 BASE-T ETHERNET INTERFACE
64
CPU
66
RAM
68
ROM
70
63
BACKPLANE INTERFACE
62
64/256-QAM DOWNSTREAM MODULATOR W/FEC
72
DATA
HEADEND MAC
60
CLOCK
DATA
CLOCK
QPSK/16-QAM BURST RECEIVER CHANNEL 1
58
SAW
74
AMP
76
SPLITTER (1:N)
57
1
N
44 MHz IF OUTPUT
RF UPCONVERTER (54-860 MHz)
78
COAX
52
OPTICAL TO COAX
79
OPTICAL
54
COAX
49
5-42 MHz RF INPUT
56
50

FIG. 16

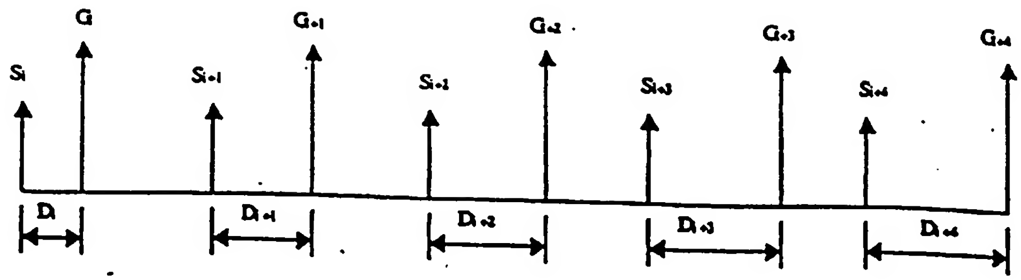


FIG. 17

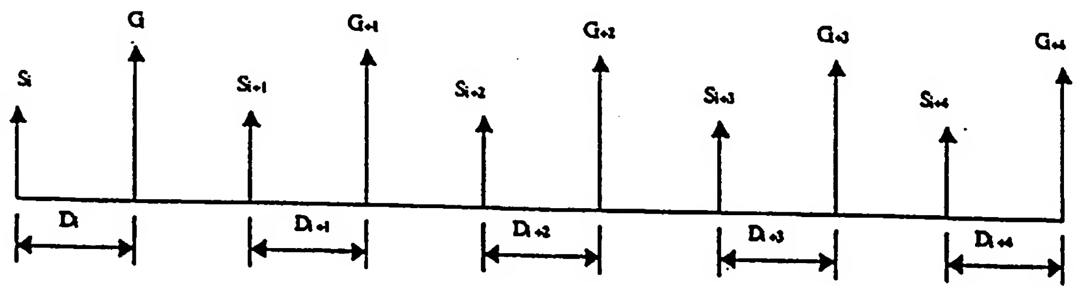


FIG. 18

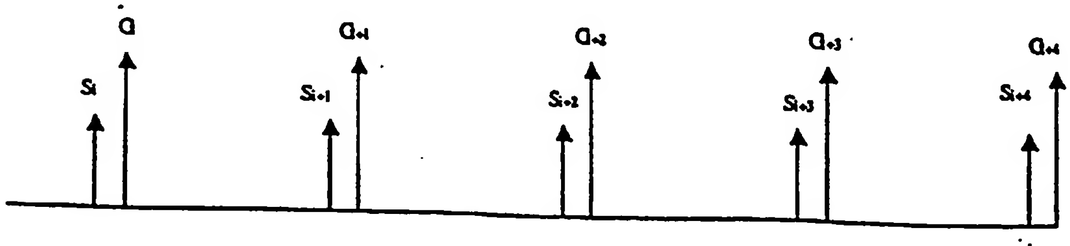


FIG. 19

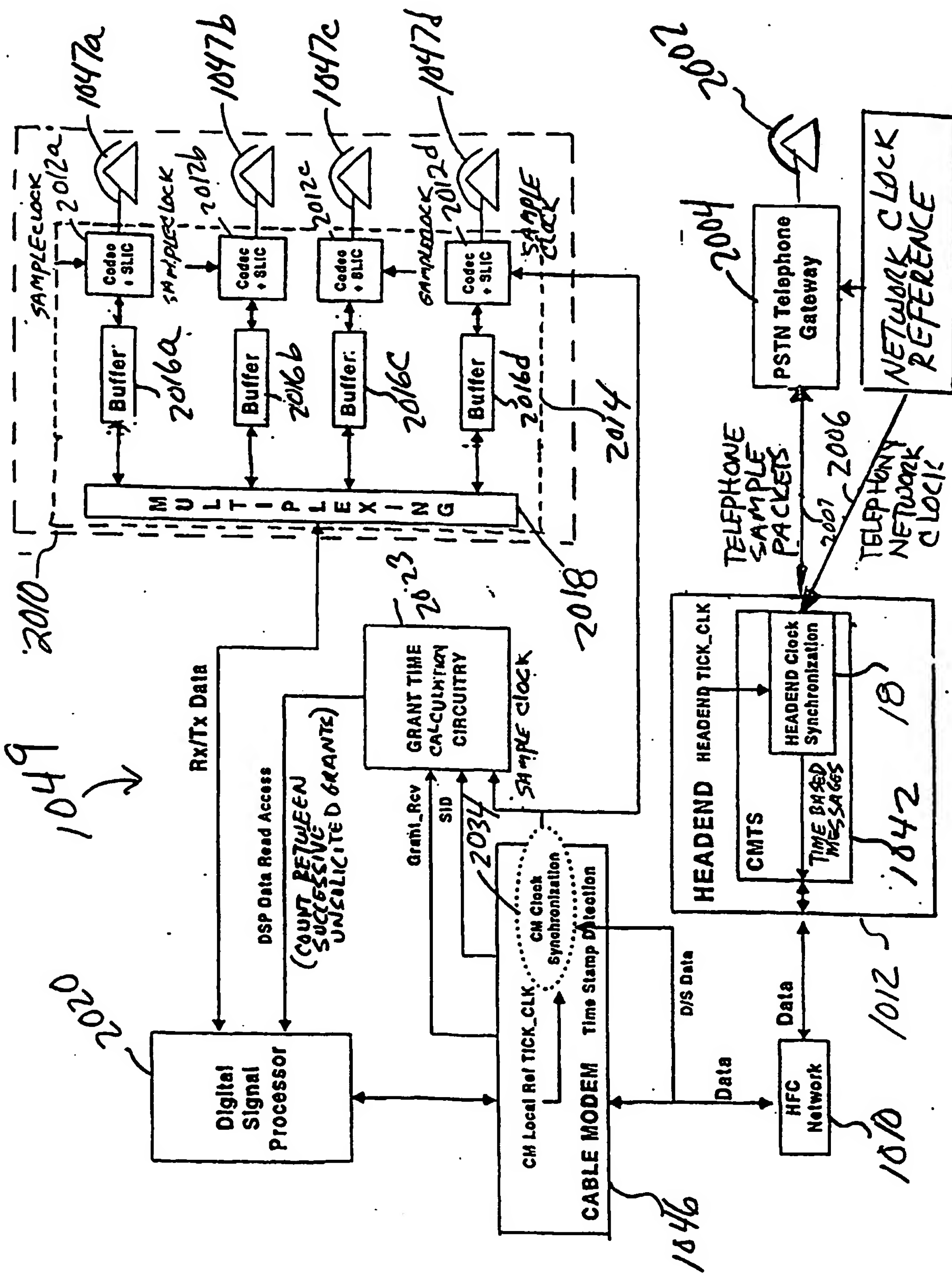

1049
2010
SAMPLE CLOCK
2012a
2012b
2012c
2012d
1047a
1047b
1047c
1047d
Codec + SLIC
Buffer
2016a
2016b
2016c
2016d
2014
MULTIPLEXING
2018
Rx/Tx Data
DSP Data Read Access
(COUNT BETWEEN SUCCESSIVE UNSOLICITED GRANTS)
2020
Digital Signal Processor
GRANT TIME CALCULATION CIRCUITRY
2023
Grant_Rcv
SID
2034
SAMPLE CLOCK
CM Clock Synchronization
CM Local Ref TICK_CLK
Time Stamp Detection
CABLE MODEM
1046
D/S Data
Data
HFC Network
1010
1012
HEADEND
CMTS
HEADEND TICK_CLK
HEADEND Clock Synchronization
18
TIME BASED MESSAGES
1842
TELEPHONE SAMPLE PACKETS
2007
2006
TELEPHONY NETWORK CLOCK
PSTN Telephone Gateway
2004
2002
NETWORK CLOCK REFERENCE
2003

FIG. 20

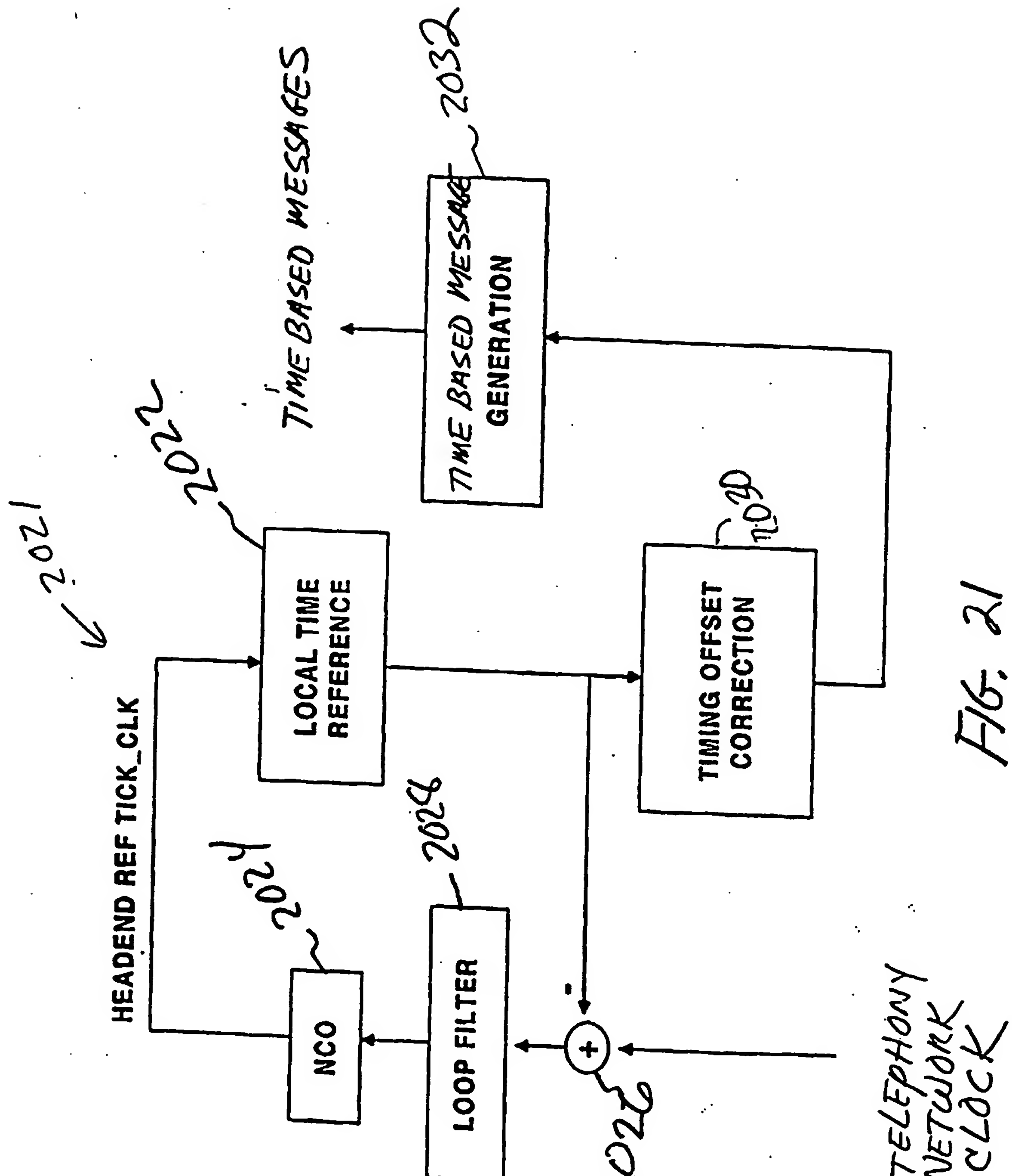

2021
HEADEND REF TICK_CLK
LOCAL TIME REFERENCE
2022
TIME BASED MESSAGES
TIME BASED MESSAGE GENERATION
2032
TIMING OFFSET CORRECTION
2030
NCO
2024
LOOP FILTER
2026
2028
TELEPHONY NETWORK CLOCK

FIG. 21

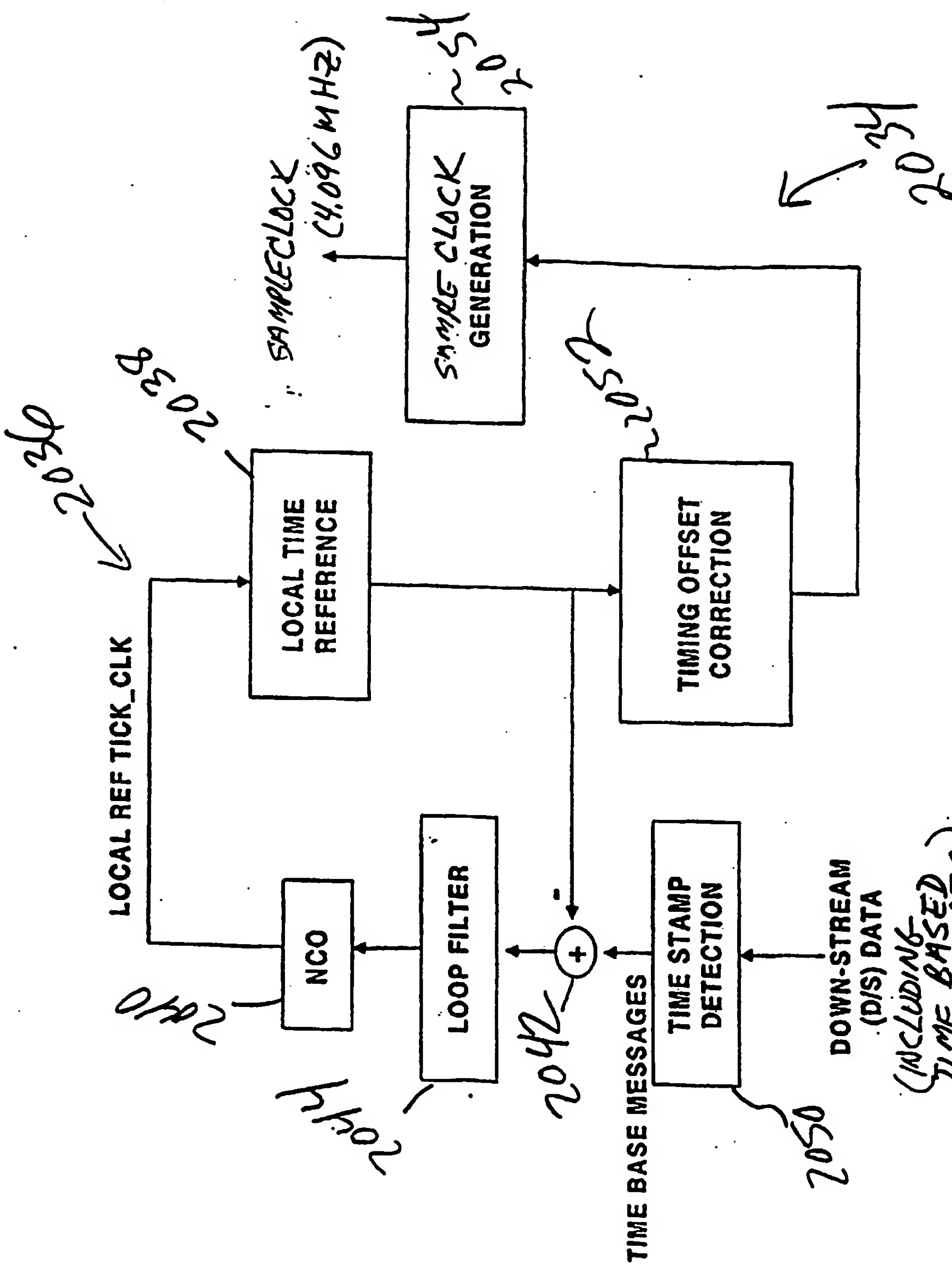

LOCAL REF TICK_CLK
2036
2038
LOCAL TIME REFERENCE
SAMPLECLOCK (4.096 MHz)
SAMPLE CLOCK GENERATION
2054
2052
TIMING OFFSET CORRECTION
2034
2040
NCO
2044
LOOP FILTER
2042
+
-
TIME BASE MESSAGES
TIME STAMP DETECTION
2050
DOWN-STREAM (D/S) DATA
(INCLUDING TIME BASED MESSAGES)

FIG. 22

FIG. 23a

FIG. 23b

FIG. 23c

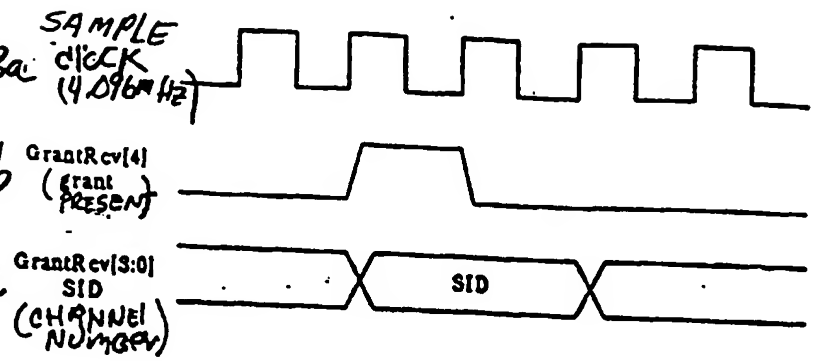
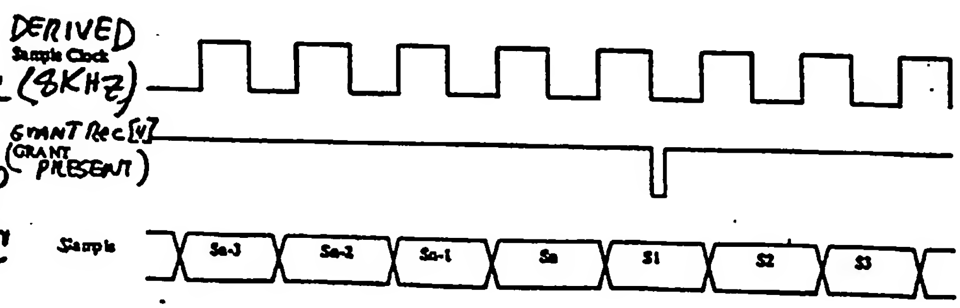


FIG. 24a

FIG. 24b

FIG 24C



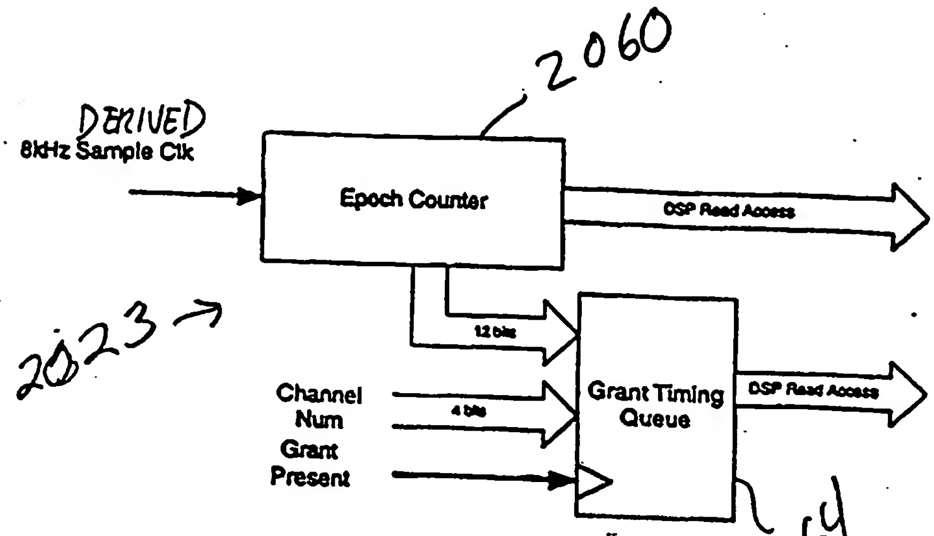

2060
DERIVED
8kHz Sample Clk
Epoch Counter
DSP Read Access
2023
12 bits
Channel
Num
4 bits
Grant
Present
Grant Timing
Queue
DSP Read Access
2064

FIG. 25

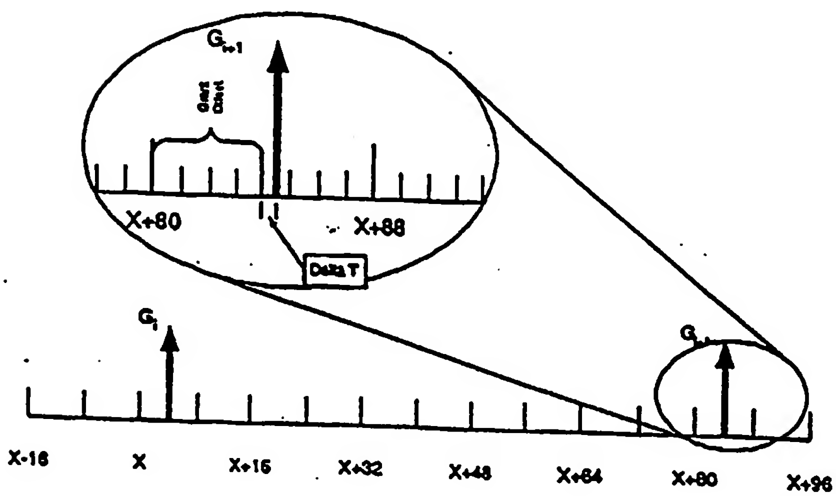

X+80
X+88
Delta T
X-16
X
X+16
X+32
X+48
X+64
X+80
X+96

FIG. 26

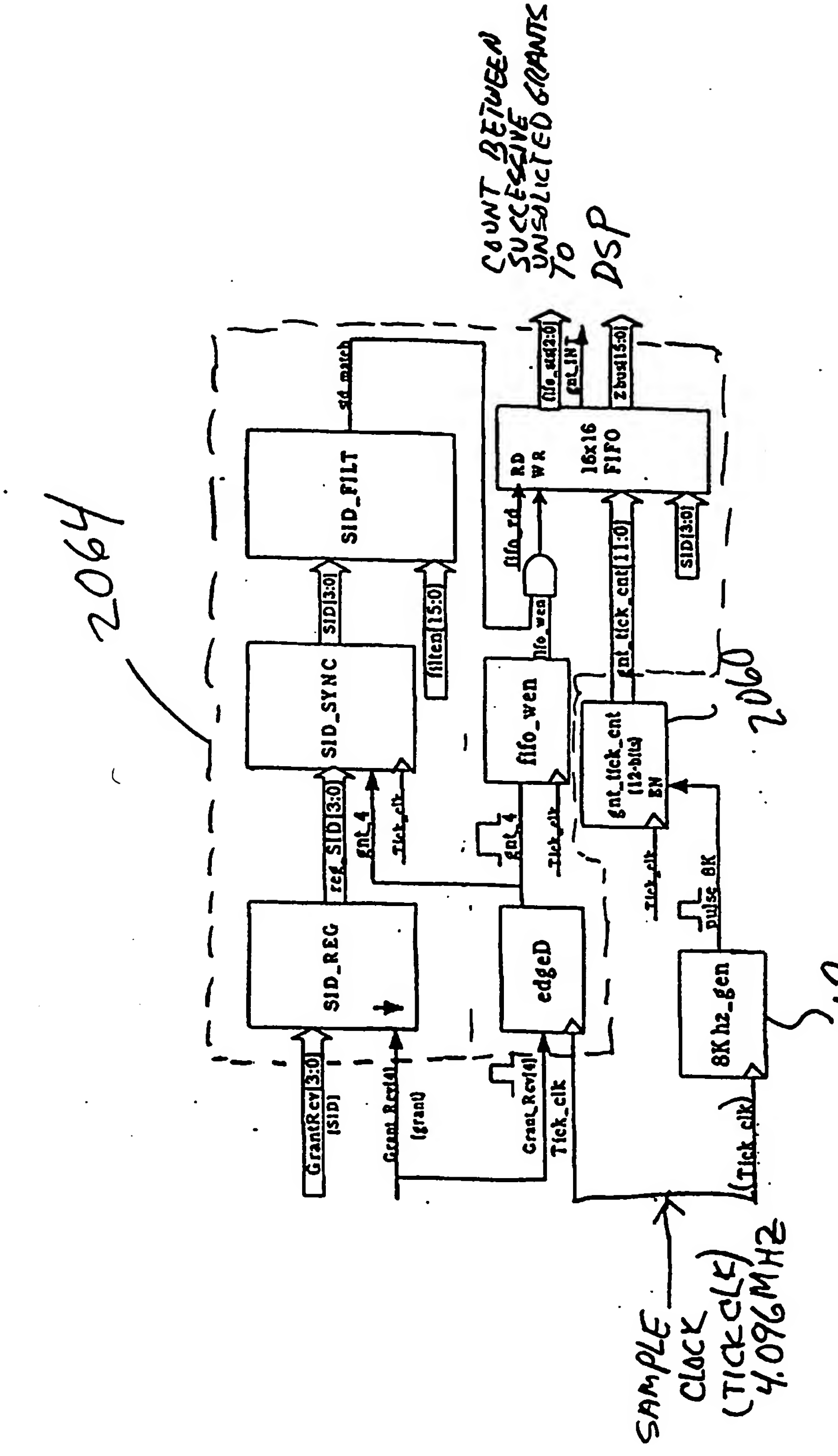

COUNT BETWEEN SUCCESSIVE UNSOLICTED GRANTS TO DSP
2064
SID_REG
SID_SYNC
SID_FILT
edgeD
fifo_wen
gnt_tick_cnt (12-bits) EN
8Khz_gen
16x16 FIFO
RD
WR
SID[3:0]
ref_SID[3:0]
gnt_4
Tick_clk
filten[15:0]
fifo_wen
fifo_rd
gnt_INT
Zbus[15:0]
gnt_tick_cnt[11:0]
pulse_8K
GrantRcv[3:0] (SID)
Grant_Rcv[4] (grant)
2060
2062
SAMPLE CLOCK (TICKCLK) 4.096MHZ

FIG. 27

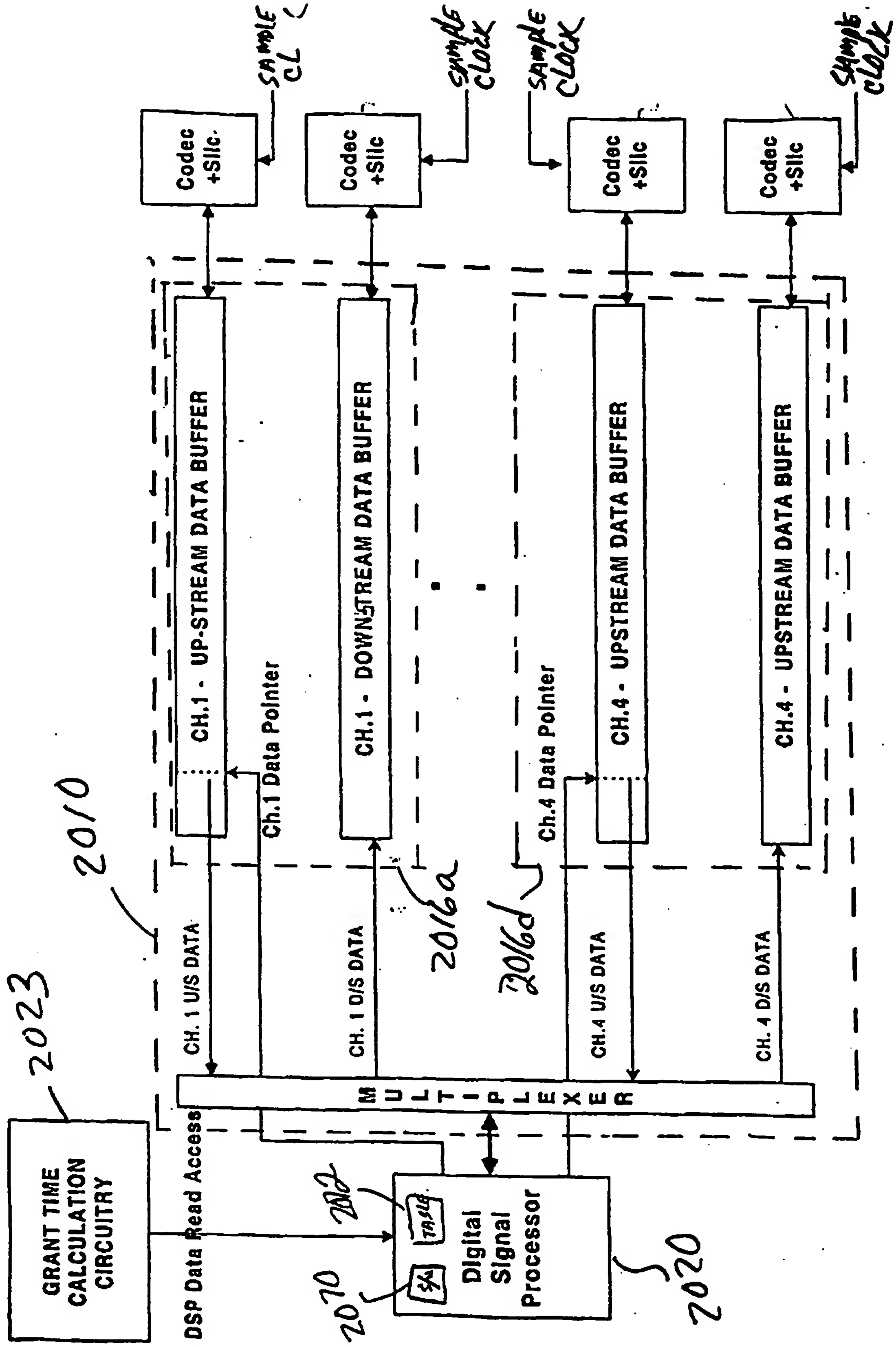

SAMPLE CLOCK
SAMPLE CLOCK
SAMPLE CLOCK
SAMPLE CLOCK
Codec +Slic
Codec +Slic
Codec +Slic
Codec +Slic
CH.1 - UP-STREAM DATA BUFFER
CH.1 - DOWNSTREAM DATA BUFFER
CH.4 - UPSTREAM DATA BUFFER
CH.4 - UPSTREAM DATA BUFFER
Ch.1 Data Pointer
Ch.4 Data Pointer
2010
2016a
2016d
2023
CH. 1 U/S DATA
CH. 1 D/S DATA
CH.4 U/S DATA
CH. 4 D/S DATA
MULTIPLEXER
GRANT TIME CALCULATION CIRCUITRY
DSP Data Read Access
2022
TASLG
SA
2070
Digital Signal Processor
2020

FIG. 28

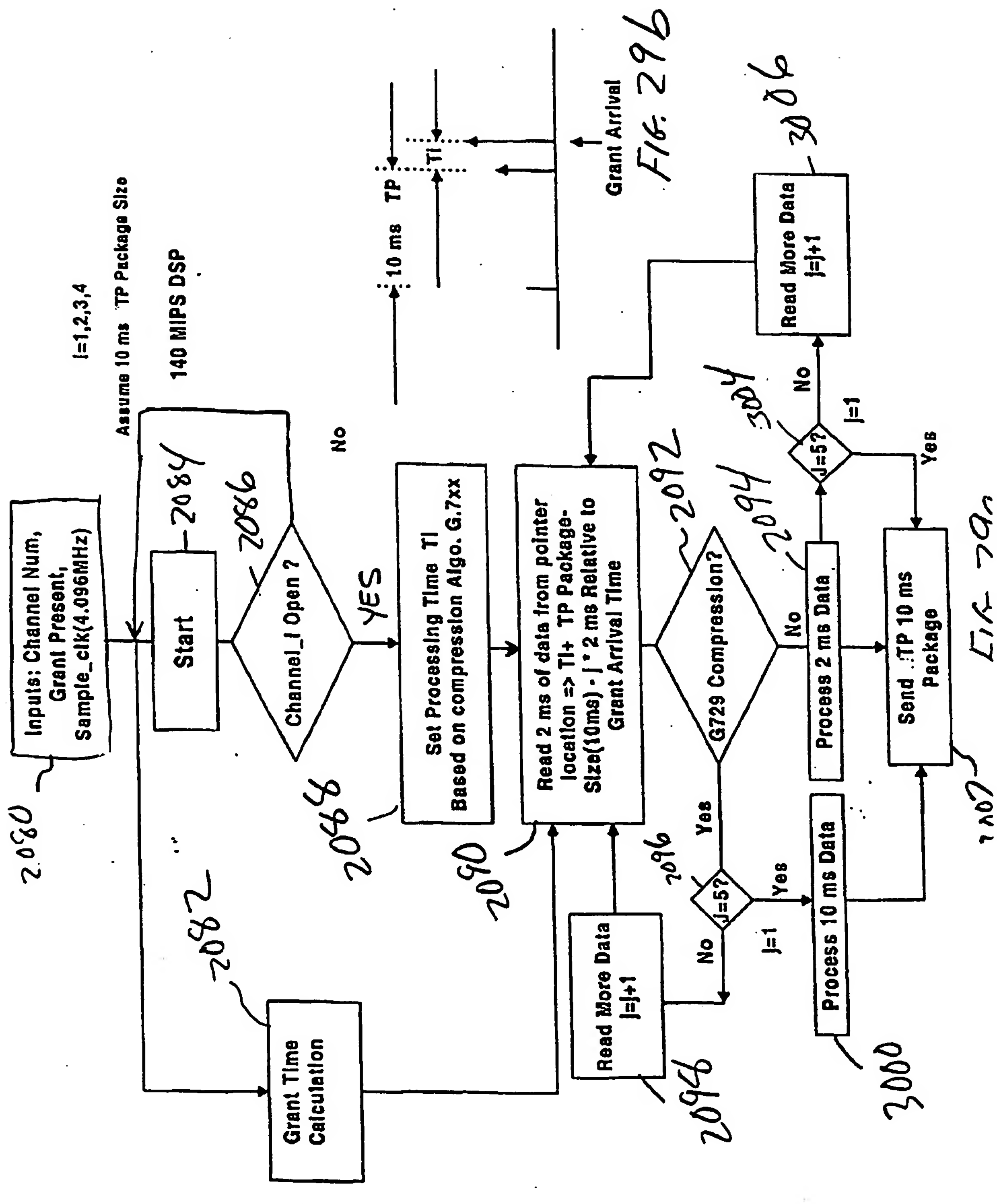


FIG. 29b

FIG. 29a

37367/CAG/B600

APPENDIX 2

SYSTEM AND METHOD FOR DELIVERING MULTIPLE VOICE CIRCUITS ON A SINGLE WIRE PAIR

System and Method for Delivering Multiple Voice Circuits on a Single Wire Pair.

An embodiment of the present invention is directed to a system and technique to deliver additional telephony and services in the home using existing wire pairs already installed in the home all while not disrupting existing services provided on the existing wire pair. FIG. 1 shows an example installation of such a system

A residential gateway may be installed at a location inside or outside the home. The residential gateway accepts inputs from an IP network on one side that is capable of delivering IP (Internet Protocol) services to the home. The other side of the residential gateway 10 can be the interface to the in home wiring that previously delivered POTS. The exemplary embodiment shown in FIG. 1 has two wire pairs, one pair continues to deliver POTS the other wire pair delivers POTS and other services to a local area network (LAN).

The residential gateway provides a means to convert the physical media and protocols used for the IP network to the physical media and protocols used on the in home wire pairs. In the described exemplary embodiment, a DOCSIS (Data Over Cable Service Interface Specification) network is used for delivery of IP services over the IP network (an HFC network). A consequence of this choice is that the residential gateway includes a cable modem.

The described exemplary embodiment uses two well-known protocols for delivery of in home services. The first protocol is a base band protocol to deliver POTS. This protocol is described by Bellcore (now Telcordia) in TR-NWT-000057. The second protocol is HomePNA (Home Phoneline Network Alliance) as described in the Version 2.0 specification.

The function of the residential gateway can be divided into three components along service delivery lines. The first is delivery of broadband data services. This function is the primary function of the cable modem as described by the CableLabs DOCSIS specification. What is unique about the residential gateway in this application is that the data service is delivered

37367/CAG/B600

using HPNA instead of Ethernet as specified by CableLabs in the DOCSIS specification.

The second function is the POTS interface. The gateway contains the high voltage circuits and the processing elements necessary to convert packetized voice delivered over IP streams to the continuous analog voltages required for the POTS interface.

The third function is a proxy for the voice over HPNA phones connected to the HPNA network. The Proxy performs an interface conversion function at two levels, first is a transport packet conversion and the second is the signaling protocol conversion.

In FIG. 1 there are two POTS phones. Both of these are traditional telephones connected to the residential gateway for telephone service. As described above, for installations where only a single wire pair is available in the home, only one phone line is used, that would be the phone attached to the HomePNA network. Not shown in this drawing is the possibility of bridging additional POTS telephones on the wire pair. In this system, these bridged phones will behave as a bridged phone on a traditional POTS line. All bridged telephones are assigned to the same phone number and the ring/dial tone behavior is as described in TR-NWT-000057.

In FIG. 1, in home appliance control is represented by a coffeepot. The concept here is to allow appliance controllers on the network to access control information for connected devices. For example, a connected personal computer might control the start time for the coffee maker.

Also shown in FIG. 1 is a connected printer device. This can be any type of computer peripheral that permits resource sharing from any of multiple personal computers or other control devices connected to the HomePNA network.

There are two additional telephone devices shown in FIG. 1 connected to the HomePNA network via a HomePNA adaptors. The adapter communicates over the HomePNA network to the HomePNA proxy function that resides within the residential gateway. The telephone and fax machines shown are standard POTS devices that could be used to receive service on the POTS connections described above. In this instance, the HomePNA adapter provides two additional phone numbers that are different from the phone numbers assigned to the two POTS lines described above.

37367/CAG/B600

The HomePNA phone shown in FIG. 1 is a telephone that integrates the function of the HomePNA adapter and the telephone. This phone looks and works just like any traditional telephone, the difference is that it uses the HPNA interface to accomplish the voice transport and signaling functions instead of the POTS interface.

The connection of five telephone devices shown in FIG. 1 allows these devices to be connected with up to five independent telephone numbers. Note that these five phone numbers are supported using only two phone wire pairs. Using traditional POTS interfaces, five phone numbers requires five wire pairs. The limit of five telephone connections is imposed for ease of description only. This method can be used to support any number of phones within the home.

FIG. 1 shows the connection of two personal computers. One shows Net Meeting and the other is described as Netscape. These describe two possible applications that are supported by personal computers connected to networks, in this case an HomePNA network. Any application can be substituted here, the important feature of these applications is that they connect to the world wide net (or Internet) through the residential gateway.

The last item shown connected to the HomePNA network in FIG. 1 is a television. This can be used to display television programming streamed from the external IP network or spooled from memory systems of an attached video server. This video server could be a dedicated device for this purpose or specialized programming on one of the attached personal computers.

1. Cable Modem

1.1 Cable Modem Architecture.

The described exemplary embodiment may provide a highly integrated solution implemented single chip that is compliant with the (DOCSIS). DOCSIS was developed to ensure that cable modem equipment built by a variety of manufacturers is compatible, as is the case with traditional dial-up modems. The described exemplary embodiment can provide integrated functions for communicating with the CMTS. For example, a QPSK upstream modulator 102 transmits data to the far end data terminating device, a QAM downstream demodulator 100 receives data from the far end data terminating device via a CMTS, and a QPSK out of band downstream demodulator 106 receives out of band MPEG-2 encoded messages from the CMTS.

37367/CAG/B600

In addition, the described exemplary embodiment can support multiple inputs in accordance with a variety of protocols. For example, a universal serial bus transceiver 104 provides transparent bi-directional IP traffic between devices operating on a USB such as for example a PC workstation, server printer or other similar devices and the far end data terminating device. Additionally, an I.E.E 802.3 compliant media independent interface (MII) 110 in conjunction with an Ethernet MAC 134 also provide bi-directional data exchange between devices such as, for example a number of PCs and or Ethernet phones and the far end data terminating device. A voice and data processor 160 is used for processing and exchanging voice, as well as fax and modem data between packet based networks and telephony devices.

The QAM downstream demodulator 100 may utilize either 64 QAM or 256 QAM in the 54 to 860 MHz bandwidth to interface with the CMTS. The QAM downstream demodulator 100 accepts an analog signal centered at the standard television IF frequencies, amplifies and digitizes the signal with an integrated programable gain amplifier and A/D converter. The digitized signal is demodulated with recovered clock and carrier timing. Matched filters and then adaptive filters remove multi-path propagation effects and narrowband co-channel interference. Soft decisions are then passed off to an ITU-T J.83 Annex A/B/C compatible decoder. The integrated decoder performs error correction and forwards the processed received data, in either parallel or serial MPEG-2 format to a DOCSIS Media Access Controller (MAC) 112.

The output of the downstream demodulator100 is coupled to the DOCSIS MAC 112. The DOCSIS MAC 112 may include baseline privacy encryption and decryption as well as robust frame acquisition and multiplexing with MPEG2-TS compliant video and audio streams. The DOCSIS MAC 112 implements the downstream portions of the DOCSIS protocol. The DOCSIS MAC 112 extracts DOCSIS MAC frames from MPEG-2 frames, processes MAC headers, and filters and processes messages and data.

Downstream data packets and message packets may be then placed in system memory 114 by a SDRAM interface 116 via the internal system bus 118. The SDRAM interface 116 preferably interfaces to a number of off the shelf SDRAMs which are provided to support the high bandwidth requirements of the Ethernet MAC 112 and other peripherals. The SDRAM interface 116 may support multiple combinations of 8, 16 or 32 bit wide SDRAMs, allowing for external data storage in the range of about 2 to 32 MBytes. The DOCSIS MAC 112 includes a number of direct memory access (DMA) channels for fast data access to and from the system memory 114 via the internal system bus 118.

The upstream modulator 102 provides an interface with the CMTS. The upstream modulator 102 may be configured to operate with numerous modulation schemes including QPSK and 16-QAM. The upstream modulator 102 supports bursts or continuous data, provides forward error correction (FEC) encoding and pre-equalization, filters and modulates the data stream and provides a direct 0-65 MHz analog output.

The DOCSIS MAC 112 can also implement the upstream portions of the DOCSIS protocol before transmission by the upstream modulator 102. The DOCSIS MAC 112 receives data from one of the DMA channels, requests bandwidth and frames the data for TDMA with other modems on the same upstream frequency.

The DOCSIS MAC interfaces with the MIPS core 128 via the ISB 118. An exemplary embodiment of the MIPS core 128 includes a high performance CPU operating at a speed of at least 80 MHz with 32-bit address and data paths. The MIPS core includes two way set associative instruction and data caches on the order of about 4kbytes each. The MIPS core 128 can provide standard EJTAG support with debug mode, run control, single step and software breakpoint instruction as well as additional optional EJTAG features.

The upstream modulator 102 and the downstream demodulator 100 are controlled by the MIPS core 128 via a serial interface which is compatible with a subset of the Motorola M-Bus and the Philips I^2C bus. The interface consists of two signals, serial data (SDA) and serial clock (SCL), which may control a plurality of devices on a common bus. The addressing of the different devices may be accomplished in accordance with an established protocol on the two wire interface.

The described exemplary embodiment of the network gateway includes a full-speed universal serial bus (USB) transceiver 1104 and USB MAC 122 which is compliant with the USB 1.1 specification. The USB MAC 122 provide concurrent operation of control, bulk, isochronous and interrupt endpoints. The USB MAC 122 also can support standard USB commands as well as class/vendor specific commands. The USB MAC 122 include integrated RAM which allows flexible configuration of the device. Two way communication of information to a device operating on a USB can be provided, such as for example a PC on a USB 1.1 compliant twisted pair. The USB MAC 122 can be arranged for hardware fragmentation of

37367/CAG/B600

higher layer packets from USB packets with automatic generation and detection of zero length USB packets. The USB MAC 122 may include DMA channels which are used to communicate received data to the system memory 114 via the internal system bus 118. Data stored in system memory 114 may then be processed and communicated to the cable modem termination system(not shown) via the DOCSIS MAC 112 and the upstream modulator 102. Similarly data received from the cable modem termination system and processed by the downstream demodulator 100 and stored in system memory as higher layer packets can be retrieved by the USB MAC122 via the ISB 118 and assembled into USB packets with automatic generation of zero length USB packets. USB packets may then be communicated to the external device operating on the USB via the USB transceiver 104.

A media independent interface (MII) 110 can provide bi-directional communication with devices such as for example a personal computer (PC) operating on an Ethernet. The media independent interface 110 can forward data to and receive information from the Ethernet MAC 134. The Ethernet MAC 134 can also perform all the physical layer interface (PHY) functions for 100BASE-TX full duplex or half-duplex Ethernet as well as 10BBASE-T full or half duplex. The Ethernet MAC 134 can also decode the received data in accordance with a variety of standards such as for example 4B5b, MLT3, and Manchester decoding. The Ethernet MAC can perform clock and data recovery, stream cipher de-scrambling, and digital adaptive equalization. The Ethernet MAC 134 may include DMA channels which are used for fast data communication of processed data to the system memory 114 via the internal system bus 118. Processed data stored in system memory 114 may then be communicated to the cable modem termination system(not shown) via the upstream modulator 102. Similarly, data received from the cable modem termination system is processed by the downstream demodulator 100 and stored in system memory as higher layer packets which can then be retrieved by the Ethernet MAC 1134 via the ISB 118 and encoded into Ethernet packets for communication to the external device operating on the Ethernet via the MII 110. The Ethernet MAC 134 may also perform additional management functions such as link integrity monitoring, etc.

In addition to the SDRAM interface 116, the described exemplary embodiment of the gateway includes a 16-bit external bus interface (EBI) 140 that supports connection to flash memories 142, external SRAM 144 or EPROMS 144. Additionally, the EBI 140 may be used to interface the described exemplary network gateway with additional external peripherals. The EBI 140 can provide a 24 bit address bus and a 16-bit bi-directional data bus. Separate read and write strobes can be provided along with multiple firmware configurable chip select signals.

37367/CAG/B600

Each chip select can be fully programmable, supporting block sizes between about 4 K-bytes and 8 M-bytes, extended clock cycle access control and 8 or 16-bit selection of peripheral data bus width. In the described embodiment, the EBI 140 can support both synchronous and asynchronous transfers. Pseudonymous transfers may be supported through the use of read/write strobes to indicate the start and duration of a transfer. The EBI 140 can include DMA access capability to or from the SDRAM interface 116. The DMA operation may take one or more forms. For example, in EBI mode, an EBI bridge can act as the DMA controller, and perform all pointer and buffer management tasks during DMA operations. In an external mode, an external device can act as the DMA controller and the EBI 140 can serve as a simple bridge. In DMA mode the MIPS core128 can be responsible for DMA setup.

The network gateway may be vulnerable to network breaches due to peripheral devices such as PC employing windows or network Macintosh computers. These operating systems include "file sharing" and "printer sharing" which allow two or more networked computers in a home or office to share files and printers. Therefore the exemplary embodiment of the gateway includes IP security module 1148 which interfaces with ISB 118. The MIPS core128 can set-up and maintain all security associations. The MIPS core128 can also filter all IP traffic and route any messages requiring security processing to the security module via the ISB 118. The security module 150 may support single DES (CBC and ECB modes) triple DES (CBC and ECB modes) MD-5 and SHA authentication in hardware to provide support for virtual private networks.

The security module 148 can implement the basic building blocks of the developing IP Security Standard (IPsec). The security module 148 may also be used to implement any other security scheme that uses the same basic blocks as IPsec, which uses two protocols to provide traffic security. A first protocol, IP Encapsulating Security Payload (ESP), provides private data privacy with encryption and limited traffic flow confidentiality. ESP may also provide connection less integrity, data source authentication and an anti-replay service. A second format, IP Authentication Header (AH), provides connection less integrity, data source authentication and an optical anti-replay service. Both protocols may be used to provide access based on the distribution of cryptographic keys and the management of traffic flows. The protocols may be used alone or in combination to satisfy the security requirements of a particular system. In addition, the security module 148 can support multiple modes of operation depending on a security association to the traffic carried by a simplex connection. For example, transport mode security association between two hosts, primarily protects protocols above the IP layer while tunnel mode security association provides security and control to a tunnel of IP packets.

37367/CAG/B600

The exemplary security module 148 addresses possible differences in packet format between IPsec and future security applications with a generalized scheme to determine where the authentication / encryption algorithms are applied with a data packet. The authentication / encryption algorithms consider each packet to consists of three parts, a header, body and trailer. The appropriate algorithm can be applied, using any specified parameters to the body section only.

In an encryption mode, the security module 148 can add and initialize any necessary headers, determine necessary parameters, generate the associated control message and add the control and data message. In the authentication mode, the control fields of the received data packets are parsed, the parameters are determined via a security association lookup table, control message is created and the control and data messages are enqueued.

The exemplary embodiment of the network gateway includes a DMA controller 150 having a number of channels that enable direct access over the ISB 118 between peripherals and the system memory 114. With the exception of the security module 148, packets received by the network gateway 98 cause DMA transfers from a peripheral to memory, which is referred to as a receive operation. A DMA transfer from memory to a peripheral is referred to as a transmit operation. Programmable features in each channel can allow DMA controller 150 to manage maximum ISB burst lengths for each channel, enable interrupts, halt operation in each channel, and save power when certain modules are not operational. The maximum ISB burst length may be programmed independently for each channel preferably up to 64 32 bit words. Each channel can include maskable interrupts connected to the MIPS core128 which indicate buffer complete, packet complete and or invalid descriptor detected. Busy DMA channels may be stalled or completely disabled by the MIPS core128. Source clocks (not shown) for each channel are can be connected to the channels based on the internal peripheral they service. For power reduction, these clocks may be turned off and on coincident with the respective peripheral's clock.

The DMA controller 150 can be operable in both non-chaining and chaining mode. In the non-chaining mode the DMA channel refers to its internal registers for the pertinent information related to a scheduled DMA burst transfer. The DMA controller can set-up the buffer start address, byte count, and status word registers before initiating the DMA channel for each allocated buffer. In the transmit direction, the DMA channels can send the specified number of bytes (preferably up to 4095) from the specified byte address. In the receive direction, the DMA channels can insert data into a specified memory location until a buffer has been

37367/CAG/B600

completely filled or the end of a packet is detected.

In the chaining mode, the system memory can be partitioned as shown in FIG. 4 preferably using descriptor rings containing pointers to memory buffers as well as status information for each memory buffer. The MIPS core128 can write the descriptor pointers while the DMA controller 150 follows by inserting/taking data into/from the location designated by the descriptor. Upon completion of the transfer of a buffer, the DMA controller 150 effectively clears the descriptor by updating the status to indicate that the data has been inserted/taken. Specific information may be added to the descriptor to indicate the length of data in the block, specifying whether the data is the first or last block of a packet, etc.

In the downstream direction, the MIPS core 128 can fill or recognize a data block for a particular DMA channel, then write the next unused descriptor in the ring indicating that the block is filled and where the downstream data exists in memory. The DMA controller 1150 can follow the DSP write to the descriptor ring, sending out data and clearing the descriptor when the transfer is complete. When the DMA controller 150 reads a descriptor that does not contain valid data, it can go idle until initiated by the DSP core.

In the upstream direction, the MIPS core128, can allocates memory space for incoming data, then write the descriptor with the start address for that buffer. The DMA controller 150 read the base address and insert data until either the buffer is full or an end of packet has been detected. The DMA controller 150 can update the descriptor, communicating to the MIPS core128 that the block is full , indicating the length of the data on the block, and/or asserted first and or last buffer flags.

The described exemplary network gateway can include a voice processor 160 for processing and transporting voice over packet based networks such as PCs running network on a USB (Universal Serial Bus) or an asynchronous serial interface, Local Area Networks (LAN) such as Ethernet, Wide Area Networks (WAN) such as Internet Protocol (IP), Frame Relay (FR), Asynchronous Transfer Mode (ATM), Public Digital Cellular Network such as TDMA (IS-13x), CDMA (IS-9x) or GSM for terrestrial wireless applications, or any other packet based system. The described embodiment of the voice processor 160 also supports the exchange of voice, as well as fax and modem, between a traditional circuit switched network or any number of telephony devices and the CMTS (not shown). The voice processor may be implemented with a variety of technologies including, by way of example, embedded communications software that

37367/CAG/B600

enables transmission of voice over packet based networks.

The exemplary embedded communications software may be implemented with the programmable DSP software architecture in combination with the MIPS core 128. Referring to FIG. 3, the embedded communications software includes a media terminal adapter (MTA) 620 comprising a host application programming interface (HAPI) 621 that provides a software messaging interface between the MIPS host and the voice and data processor DSP. The HAPI 621 facilitates the issuing of commands from the MIPS host to the voice and data processor DSP as well the sending of events from the DSP to the MIPS core host.

In addition, the MTA 620 can provide all signaling and encapsulation elements required to provide telephony service over a DOCSIS HFC network 622 including media transport and call signaling via quality service logic 623. For example, gateway control protocol (GCP) logic 624 receives and mediates call-signaling information between the PacketCable network and the PSTN. The GCP logic 624 maintains and controls the overall call state for calls requiring PSTN interconnection. The GCP logic 624 controls the voice and data processor 626, via the MTA 620 and HAPI interface 621, by instructing it to create, modify, and delete connections that support the media stream over the IP network. The GCP logic 624 also instructs the voice and data processor to detect and generate events and signals. The GCP logic 624 also exercise attribute control over the voice and data processor 626 providing instructions as to which attributes to apply to a connection, such as, for example, encoding method, use of echo cancellation, security parameters, etc.

The GCP logic 624 also interfaces with an external control element called a call agent or call management server (CMS) 628 to terminate and generate the call signaling from and to the PacketCable side of the network in accordance with the network-based call signaling (NCS) protocol specification. The PacketCable 1.0 NCS architecture places call state and feature implementation in the centralized CMS 628, and places telephony device controls in the MTA 620. The MTA 620 passes device events to the CMS 628, and responds to commands issued from the CMS. The CMS, is responsible for setting up and tearing down calls, providing advanced services such as custom calling features, performing call authorization, and generating billing event records, etc. For example, the CMS 628 instructs the MTA 620 to inform the CMS 628 when the phone goes off hook, and seven dual tone multi frequency (DTMF) digits have been entered. The CMS 628 instructs the MTA 620 to create a connection, reserve quality of service (QoS) resources through the access network for the pending voice connection, and to play

a locally generated ringback tone. The CMS in turn communicates with a remote CMS (or MGC) to setup the call. When the CMS detects answer from the far end, it instructs the MTA to stop the ringback tone, activate the media connection between the MTA and the far-end MTA, and begin sending and receiving media stream packets.

When a voice channel is successfully established, real time transport protocol (RTP) is used to transport all media streams in a PacketCable compliant network to guarantee interoperability. Real time transport protocol (RTP) provides end-to-end delivery services for data with real time characteristics, such as interactive audio and video. Those services include payload type identification, sequence numbering, timestamping and delivery monitoring of the quality of service (QoS) and conveys to participants statistics such as for example packet and byte counts for the session. RTP resides right above the transport layer. The described exemplary embedded MTA 620 preferably includes RTP logic 630 that converts RTP packets (headers) to a protocol independent format utilized by the voice processor 626 and vice versa.

The described exemplary embedded MTA preferably includes channel associated signaling (CAS) logic 632 resident on the MIPS core that interfaces with the subscriber line interface circuits 634 via the GPIO interface 184 (see FIG. 3) to provide ring generation, hookswitch detection, and battery voltage control. The CAS logic 632 preferably supports custom calling features such as for exam distinctive ringing.

The described exemplary embedded MTA 620 preferably includes MTA device provisioning logic 636 which enables the embedded MTA 620 to register and provide subscriber services over the HFC network 622. Provisioning logic 636 provides initialization, authentication, and registration functions. The Provisioning logic 636 also provides attribute definitions required in the MTA configuration file. The provisioning logic 636 includes a SNMP logic 638 that exchanges device information and endpoint information between the MTA 620 and an external control element called a provisioning server (not shown). The MTA also sends notification to the provisioning server that provisioning has been completed along with a pass/fail status using the SNMP protocol.

The Provisioning logic 636 also includes DHCP logic 640 which interfaces with an external dynamic host configuration protocol (DHCP) server to assign an IP address to the MTA. The DHCP server (not shown) is a back office network element used during the MTA device provisioning process to dynamically allocate IP addresses and other client configuration

37367/CAG/B600

information. Further provisioning logic preferably includes domain name server (DNS) logic 642 which interfaces with an external DNS server(not shown) to obtain the IP address of a PacketCable server given its fully\qualified domain name.

The MTA configuration file is downloaded to the MTA from an external trivial file transfer protocol (TFTP) server (not shown) through TFTP logic 644. The TFTP server is a back office network element used during the MTA device provisioning process to download configuration files to the MTA. An HTTP Server may be used instead of a TFTP server to download configuration files to the MTA.

Each of PacketCable's protocol interfaces is subject to threats that could pose security risks to both the subscriber and service provider. The PacketCable architecture addresses these threats by specifying, for each defined protocol interface, the underlying security mechanisms (such as IPSec) that provide the protocol interface with the security services it requires, e.g., authentication, integrity, confidentiality. Security logic 646 is PacketCable compliant and provides for voice and provides end-to-end encryption of RTP media streams and signaling messages, to reduce the threat of unauthorized interception of communications. The security logic 646 preferably provides additional security services such as, for example, authentication, access control, integrity, confidentiality and non-repudiation.

DOCSIS service logic 648 preferably provides the primary interface between the MTA 620 and the DOCSIS cable modem (i.e. DOCSIS MAC and modulator / demodulator) of the network gateway. The DOCIS service logic 648 provides multiple sub-interfaces such as for example a control sub-interface which manages DOCSIS service-flows and associated QoS traffic parameters and classification rules as well as a synchronization interface which is used to synchronize packet and scheduling prioritization for minimization of latency and jitter with guaranteed minimum constant bit rate scheduling. In addition, the DOCSIS service logic is used to request bandwidth and QoS resources related to the bandwidth. The DOCIS cable modem features of the network gateway then negotiate reserve bandwidth, guaranteed minimum bit rate etc, utilizing DOSCIS 1.1 quality of service feature. Similarly, DOCSIS service logic 648 preferably includes a transport interface which is used to process packets in the media stream and perform appropriate per-packet QoS processing.

37367/CAG/B600

The exemplary embedded MTA may best be illustrated in the context of a typical voice communication across the DOCSIS HFC network. The user initiates a communication by going off hook. The CAS detects the off hook condition from the SLIC and sends an off hook event to the MTA call client. The MTA call client then instructs the GCP logic to generate a off hook signal. The GCP logic generates an of hook signal which is forwarded to the MTA call client and transmitted out the QoS service logic to the call management server via the DOCSIS MAC and upstream modulator of the network gateway and the CMTS. The call management server typically would transmit a return signal via the CMTS, DOCSIS MAC and downstream demodulator of the network gateway to the MTA call client via the QoS service logic. The MTA call client preferably forwards that signal to the GCP logic which decodes the signal, typically play dial tone. The GCP logic would then signal the MTA call client to play dial tone. The MTA call client then sends a command to the voice processor via the HAPI interface to play dial tone. The user then hears a dial tone.

Upon hearing a dial tone a user will then typically dial a number. The voice processor includes a DTMF detector which detects the dialed digits and forwards the detected digits to the MTA call client as events via the HAPI interface. The MTA call client forwards the event to the GCP logic which encodes the dialed digits into a signaling message which is returned to the MTA call client. The MTA call client transmits the signaling message out the QoS service logic to the call management server via the DOCSIS MAC and upstream modulator of the network gateway and the CMTS. The call management server would then instruct a called party MTA to generate a ring to the called number. If the called number answers by going off hook, the CAS of the called MTA would detect an off hook condition and signal the call management server. The call management server then instructs the MTA call client via the CMTS, and downstream demodulator, DOCSIS MAC and QoS service logic of the network gateway to establish a voice connection with a given set of features, i.e. use echo cancellation, and silence suppression, use given coder etc. In addition, the MTA call client is given the IP address of the called party, to which the RTP voice packets should be sent. The MTA call client forwards the received message to the GCP logic which decodes the received message. The GCP logic generates attribute instructions for the voice processor such as, for example, encoding method, use of echo cancellation, security parameters, etc. which are communicated to the voice processor via the MTA call client and the HAPI interface.

Voice packets are then exchanged. For example, if the calling party speaks, the voice processor would processor the voice and forward voice packets the MTA call client via the HAPI

37367/CAG/B600

interface. The MTA call client would then forward those voice packet the RTP logic which would convert the packet from a protocol independent packet format to the RTP format. The RTP voice packets are then returned to the MTA which transmits the RTP voice packet to the CMTS via the QoS service logic and the DOCSIS MAC and upstream demodulator of the network gateway. The voice packets are then routed to the called party. Similarly, voice packets from the called party are communicated to the MTA of the call client via the QoS service logic. The MTA call client forwards the RTP voice packets to the RTP logic which converts the packet from the RTP format to the protocol independent packet format. The protocol independent voice packets are returned to the MTA call client which forwards them to the voice processor via the HAPI interface. The voice processor decodes the packets and communicates a digital stream to the called party. Voice exchange would continue in a similar manner until an on hook condition is detected by either the calling or called party CAS which would forwarded a on hook detection event to its respective MTA. The MTA would instructs the GCP logic to generate a hook detection signaling message which is returned to the MTA and forwarded to the call management server. The call management server would generate a request to play (dial tone, silence or receiver off hook) which is forwarded to the opposite MTA. The MTA would forward the request to the GCP logic which would then instruct the voice processor to play dial tone via the MTA and HAPI interface.

Telephony calls in the other direction are similarly processed. For example, the call management server instructs the MTA called client to ring a dialed number. The MTA called client instructs the GCP logic to generates an command to ring the dialed number. The command is then forwarded to the CAS via the MTA called client. The CAS generates a ring signal and forwards that signal to the SLIC which then rings the called telephony device. The MTA called client may also instruct the GCP logic to present call ID which preferably generates a command for the voice processor to present caller ID. If the user picks up the phone the CAS would detect an off hook condition and signal an off hook event back to the MTA. The MTA called client would then instruct the GCP logic to create an off hook detection signaling message, which when created is returned to the MTA and forwarded to the external call management server via the QoS service logic, DOCSIS MAC and upstream modulator of the network gateway and the CMTS. A communication channel would again be established with a given set of attributes as previously described. The embedded communications software is preferably run on a programmable digital signal processor (DSP). In an exemplary embodiment the voice processor 160 utilizes a ZSP core from LSI Logic Core ware library for mid to high end telecommunications applications. The DSP core 160 can include at least about 80k words internal instruction RAM 162 and at least

about 48k words internal data RAM 164. The DSP core 160 can interface with the internal system bus 118 via a DSP/ISB interface 166 and the peripheral bus 132 via the DSP/PB interface 168.

The voice processor software enables transmission of voice, fax and data packets over packet based networks. The voice processor includes a voice exchange between a telephony device and the DOCSIS network. The voice exchange provides numerous functions including, by way of example, echo cancellation to remove far end echos, DTMF detection, voice compression/decompression algorithms, jitter buffering to compensate for network jitter, lost frame recovery, and comfort noise generation during silent periods.

The voice processor may also include a fax image data relay between a standard Group 3 fax session and the DOCSIS network. The fax relay provides increased bandwidth performance over traditional voiceband fax transmissions by invoking demodulation/modulation algorithms. The fax relay may also includes spoofing techniques during rate negotiation to avoid timeout constraints.

The voice processor may also include a modem data relay between an analog line connection and the DOCSIS network. The modem relay provides increased bandwidth performance over traditional voiceband modem transmissions by invoking demodulation/modulation algorithms. The modem relay may also includes spoofing techniques during rate negotiation to avoid timeout constraints. The described exemplary embodiment of the embedded software for the voice processor is identical to that described in detail in Section 2.3.1 herein.

The DSP core 160 can provide a JTAG Emulator interface as well as internal training recovery clock (TRC) sync interface. The voice processor 160 can include a grant synchronizer that insures timely delivery of voice signals to the MIPS core 128 for upstream transmission. In addition, a PCM interface 170 can provide the voice processor 160 with an interface to an internal audio processor 170 as well as an external audio processing circuits to support constant bit rate (CBR) services such as telephony. The PCM interface can provide multiple PCM channel controllers to support multiple voice channels. In the described exemplary embodiment of the gateway, there are four sets of transmit and receive FIFO registers, one for each of the four PCM controllers. However, the actual number of channels that may be processed may vary and is limited only by the performance of the DSP. The internal system bus 118 is used to transfer

37367/CAG/B600

data, control and status messages between the voice processor 160 and the MIPS core 128. FIFO registers are preferably used in each direction to store data packets.

The described exemplary embodiment of the gateway includes an internal audio processor 170 with an analog front end 172 which interface the voice processor 169 with external subscriber line interface circuits (SLICs) for bi-directional exchange of voice signals. The audio processor 170 may include programable elements that implement filters and other interface components for a plurality of voice channels. In the transmit mode the analog front end 172 accepts an analog voice signal and digitizes the signal and forwards the digitized signal to the audio processor 170.

In the described exemplary embodiment, the audio processor 170 may include A-law / μ-law (G.711 compatible) encoder and decoder decimate the digitized signal and condition the decimated signal to remove far end echos.

As the name implies, echos in telephone systems is the return of the talker's voice resulting from the operation of the hybrid with its two-four wire conversion. If there is low end-to-end delay, echo from the far end is equivalent to side-tone (echo from the near-end), and therefore, not a problem. Side-tone gives users feedback as to how loud they are talking, and indeed, without side-tone, users tend to talk too loud. However, far end echo delays of more than about 10 to 30 msec significantly degrade the voice quality and are a major annoyance to the user. The audio processor can apply a fixed gain / attenuation to the conditioned signal and forwards the gain adjusted signal to the voice processor 160 via the PCM interface. In the receive mode the audio processor accepts a voice signal from the PCM interface and preferably applies a fixed gain/attenuation to the received signal. The gain adjusted signal is then interpolated from 8kHz to 96 kHz before being D/A converted for communication via a SLIC interface to a telephony device.

Each audio channel can be routed to a PCM port to allow for system level PCM testing. The PCM system tests, by way of example may require compliance with ITU G.711 for A-law and μ-law encoding / decoding.

The described exemplary embodiment of the network gateway include integrated peripherals including independent periodic interval timers 180, a dual universal asynchronous receiver-transmitter (UART) 182 that handles asynchronous serial communication, a number of

internal interrupt sources 184, and a GPIO module 186 that provides multiple individually configurable input/output ports. In addition, multiple GPIO ports can be provided to drive various light emitting diodes (LEDs) and to control a number of external SLICs. A peripheral bus bridge 186 can be used to interface the low speed peripheral to the internal system bus 118.

The described exemplary embodiment also includes an HPNA MAC (not shown) which provides an interface between the HomePNa network and the MIPS processor.

1.2. Cable Modem Flow Path.

FIG. 3 presents a data flow diagram that describes the flow of transport packets in the residential gateway described in FIG. 2.

The DOCSIS Interface is the primary interface to the DOCSIS network within the residential gateway. All packets that arrive to or leave from the residential gateway via the DOCSIS network must go through the DOCSIS interface block. As shown in FIG. 3, all packets arriving from the DOCSIS network go through the DOCSIS interface block and are delivered to the DOCSIS packet filter. The DOCSIS interface block translates the packet format as represented in the DOCSIS network to an internal format that is used for all packet filter and routing functions within the residential gateway.

The DOCSIS packet filter accepts packets from the DOCSIS interface and makes a routing decision based on the destination address within the packet. The destination of the packet will be one of three possibilities: (1) VoIP Packets for the proxy gateway, (2) VoIP packets for the telephony interface controller or (3) data packets delivered directly to the HPNA interface.

The HPNA interface is the primary interface to the HomePNA network within the residential gateway. All packets that arrive to or leave from the residential gateway via the HomePNA network must go through the HPNA interface block. As shown in FIG. 3, all packets arriving from the HomePNA network go through the HPNA interface block and are delivered to the HPNA packet filter. The HPNA interface block translates the packet format as represented in the HPNA network to an internal format that is used for all packet filter and routing functions within the residential gateway.

37367/CAG/B600

The HPNA packet filter accepts packets from the HPNA interface and makes a routing decision based on the destination address within the packet. The destination of the packet will be one of two possibilities: (1) VoHN packets for the proxy gateway, or (2) data packets delivered directly to the DOCSIS interface.

The proxy gateway performs a translation function between the packets in the VoHN format to packets in the VoIP format. The specific translation is direction dependent. Packets arriving from the HPNA packet filter are translated to a VoIP format and delivered to the DOCSIS interface. Packets arriving from the DOCSIS packet filter are translated to a VoHN format and delivered to the HPNA interface.

The telephone interface controller performs a media and protocol translation between VoIP formats to PCM audio samples that are delivered to the CODEC interface. This transformation may include conversion from compressed audio formats as well as signaling transformations.

The CODEC converts the PCM Sample stream to an analog audio signal delivered to the SLICs. The SLIC performs a voltage level conversion delivering the voltage levels required by the POTS interface to be delivered to the telephone equipment attached to the SLIC.

2. HomePNA Phone

FIG. 5 shows an exemplary HomePNA phone and a functional block diagram. The HomePNA phone 900 has high density packaging with a light weight construction for home and portable applications. The HomePNA 900 is shown with an exterior housing 901 formed of a suitably sturdy material and includes a dialing device such as a keypad 906. However, those skilled in the art will appreciate that various other types of dialing devices, e.g., touchpads, voice control, etc., are likewise suitable. A headset 902 is positioned over an internal speaker 904. The internal speaker is optionally part of the HomePNA phone. An LCD housing 909 is hinged to the top of the HomePNA phone 900. The LCD housing 909 may be may be opened to expose an LCD display 910 and special function keys 908.

The keypad 906 is used to enter user inputs such as telephone numbers and passwords.

The special function keys 908 are used to enter control command inputs, establish communications and to select different modes of operation. The LCD display 910 can provide the user with various forms of information including the dialed number, as well as any other desired information such as network status, caller identification, etc.

The keypad 906' is coupled to the voice engine 12 for packetizing. The handset 902' is also coupled to the voice engine 912. The handset includes a transmitter (not shown) and a receiver (not shown). The transmitter is used to couple the near end user's voice to the voice engine 912 for voice compression and packetization. The packetized compressed voice is then coupled through the HomePNA port 914 to the HomePNA network (not shown). The receiver includes a speaker (not shown) which allows the near end user to receive voice communications from a far end user.

The voice communications from the far end user are inputted from the HomePNA network (not shown) through the HomePNA port 914 to the voice engine 912. The voice engine 912 depacketizes and decompresses the voice communications and couples the voice communications to the speaker in the receiver as analog voice signals.

The voice engine 912 also controls the LCD display 916 through a serial port interface bus 922. External memory 918 may also be provided through an external bus interface 920.

The architecture for an exemplary embodiment of the voice engine is shown in FIG. 6. The voice engine includes an HPNA analog front end (AFE) 1000 for connection to the existing wire pairs in the home. The HPNA AFE 1000 provides modulation of voice packets from an external telephony device 1002 to the in home wire pairs. The HPNA AFE 1000 also provides demodulation of voice packets from the in home wire pairs for further processing before delivery to the external telephony device 1002. The HPNA AFE 1000 can be implemented in a variety of technologies including, by way of example, an integrated circuit. An exemplary integrated circuit for the HPNA AFE 1000 is described in Section 2.1 herein.

The HPNA AFE 1000 is coupled to the HPNA MAC 1004. The HPNA MAC 1004 provides the framing and link control protocol for the voice packets exchanged between the external telephony device 1002 and the in home wire pairs. The HPNA MAC 1004 can be

implemented in a variety of technologies including, by way of example, an integrated circuit. An exemplary integrated circuit for the HPNA MAC 1004 is described in Section 2.2 herein.

The HPNA MAC 1004 interfaces with a voice processor 1006 over a data bus 1007. The voice processor 1006 can be a ZSP DSP core with embedded communications software or any other technology known in the art. The described embodiment of the voice processor 1006 supports the exchange of voice, as well as fax and modem, between the single in home wire pair and the external telephony device 1002. The voice processor may be implemented with a variety of technologies including, by way of example, embedded communications software. A packet synchronizer 1012 synchronizes the processing of voice packets in the voice processor 1006 under control of the HPNA MAC 1004.

The embedded communications software enables transmission of voice, fax and data packets over packet based networks. The embedded software includes a voice exchange between a telephony device and the in home wire pair. The voice exchange provides numerous functions including, by way of example, echo cancellation to remove far end echos, DTMF detection, voice compression/decompression algorithms, jitter buffering to compensate for network jitter, lost frame recovery, and comfort noise generation during silent periods.

The embedded software may also include a fax image data relay between a standard Group 3 fax session and the in home wire pair. The fax relay provides increased bandwidth performance over traditional voiceband fax transmissions by invoking demodulation/modulation algorithms. The fax relay may also includes spoofing techniques during rate negotiation to avoid timeout constraints.

The embedded software may also include a modem data relay between an analog line connection and the in home wire pair. The modem relay provides increased bandwidth performance over traditional voiceband modem transmissions by invoking demodulation/modulation algorithms. The modem relay may also includes spoofing techniques during rate negotiation to avoid timeout constraints. The details of the described exemplary embodiment of the embedded software are discussed in Section 2.3 herein.

The voice processor 1006 is coupled to a CODED(coder-decoder) 1008. The CODEC 1008 includes an analog-to-digital converter (ADC) for digitizing voice from the external telephony device 1002 and a digital-to-analog converter (DAC) for reconstructing voice prior to delivery to the external telephony device 1002. The CODEC includes a bandlimiting filter for the ADC and a reconstruction smoothing filter for the output of the DAC. A sample synchronizer 1014 synchronizes the sampling rates of the DAC and ADC under control of the HPNA MAC 1004. Exemplary embodiments of the sample synchronizer 1014 and the packet synchronizer are described in more detail in Section 2.4 herein.

A keypad scanner 1016 provides an interface between the keypad and the voice processor 1006. The LCD interface 1018 provides an interface between LCD display and the voice processor 1006.

BROADCOM.

Voice over HomePNA Networks

Service Definition and Protocol Description

Revision 0.4

Contents

1 Introduction

This document proposes a reference service model and describes a protocol for distributing Plain Old Telephone Service (POTS) over HomePNA networks. It is not yet intended as a formal protocol specification, but as a work-in-progress of how such a protocol might operate.

1.1 Motivation

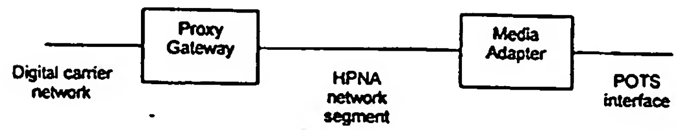
High-speed networking services to the home using HFC cable or DSL technologies are rapidly being deployed. Service operators are planning initiatives to provide multi-line POTS service as a competitive offering to the established local carrier. However, most homes are not wired to provide multi-line service, but are wired as a single line with multiple shared taps or as a star topology. A method to flexibly distribute multiple POTS lines in the home over existing wiring is highly desirable.

1.2 Scope

HomePNA is a technology that enables a 4-32Mbits/s LAN using existing in-home telephone wiring. This document proposes a method to multiplex multiple POTS service terminations as a packetized voice and signaling service over a HomePNA network. It describes the reference model, and network service description and defines the elements of procedure and formats of frames.

2 Reference Model and Service Description

The reference model is shown in Figure 1.



The service model consists of:

- An upstream carrier network service (e.g. PacketCable or GR-303 over DOCSIS, or ADSL) that terminates in a Proxy Gateway.
- A Proxy Gateway that acts as a proxy and translates between the upstream telephony service protocol and the protocol defined in this document. The upstream telephony service terminates one or more residential line services at the Proxy Gateway.
- A HomePNA network segment that provides a multiple access shared network with necessary QOS (bandwidth, reliability, timing synchronization and bounded delay characteristics) for the transport of packetized voice and call signaling between Proxy Gateways and Media Adapters.
- A Media Adapter that provides a subscriber-side interface equivalent to the standard analog phone interface defined by BellCore for residential line service using loop-start signaling, and a network-side interface defined by HomePNA 2.0 and the protocol in this document.

A goal of this service model is to permit a range of implementations of the Media Adapter device, from a "black phone" connected to an RJ-11 port on a Media Adapter "dongle", to a multi-line digital handset with integrated HPNA interface. The Media Adapter and protocol must support the use of fax machine, caller-id display, data modem, or answering machine, as well as standard voice service.

2.1 Service Overview

The service provided here is intended to operate over a single HomePNA 2.x network segment. The HomePNA network must not be shared with 1.x HPNA stations or 2.0 HPNA stations operating in 1M8 or V1M2 modes, due to the voice QOS delay requirements.

The HomePNA network may be shared with other data devices, such as PCs or printers. The necessary QOS for POTS is guaranteed through use of the services provided by the HPNA 2.0 priority-based DFPQ MAC protocol.

There may be multiple Media Adapters attached to the HomePNA network. Each Media Adapter may assigned to one or more POTS line terminations through a dynamic line binding procedures described in section 4.12. The number of POTS line terminations supported by the system is limited by the number of concurrent calls supported by the HomePNA media at the specified QOS, but is not less than 4.

There may be more than one Media Adapter that is bound to the same POTS line termination. The Proxy Gateway may choose, as an implementation decision, to exchange call signaling with all Media Adapters bound to the same POTS line termination. This allows, for example, incoming calls to ring at more than one phone, and be answered at any one, or for an outgoing call on a specific POTS line to originate at different devices.

Additional enhanced service offerings are possible to construct using the services supported by this protocol and future enhanced capability of Proxy Gateways or Media Adapters. The procedures to implement these services are not explicitly defined in the current version. Such offerings could include:

- Conference Bridging

  This would allow multiple Media Adapters bridged on to the same call, without additional carrier network resources. An example is two subscribers picking up the same call on two different handsets. Mixing of audio paths would occur in the Proxy Gateway, at the expense of some additional delay.

- Multi-line Conferencing

  This would allow a single Media Adapter to be a member of a network-hosted conference, where each party is represented by a distinct voice service flow delivered from the carrier network. Mixing of downstream audio paths would occur at the Media Adapter.

- In-home Intercom

  Station-to-station intercom could be hosted by the Proxy Gateway, without consuming any carrier network resources.

- Temporary House-Guest Line

  The multi-line support and shared access media make it easy for a service operator to provision temporary lines with separate directory numbers.

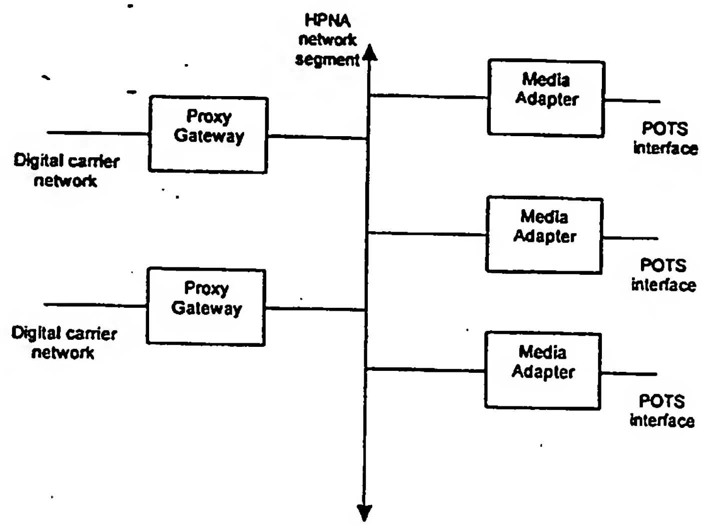
- Internal Call-Transfer, Forward

  Calls could be internally transferred from handset to handset, regardless of specific line ID, through management by the Proxy Gateway, without carrier network involvement. Advanced call features, such as call-forward-busy, call-forward-no-answer, etc to other in-home lines are also possible without carrier network involvement.

- Multiple Gateways

  Multiple Proxy Gateways could be attached to the same HomePNA network. An example might be a customer who has both residential-use POTS line termination provided by a cable operator, and a business office PBX line termination provided by ADSL. Each Proxy Gateway is responsible for a distinct non-overlapping set of POTS line terminations. The binding of POTS line termination on a Media Adapter to associated Proxy Gateway is made through the dynamic line binding procedure described in section 4.12.

  Figure 2 expands the reference model to show multiple Proxy Gateways and Media Adapters.



2.2 Functional Decomposition

The partitioning of functionality between the Media Adapter and Proxy Gateway has been strongly influenced by the packetized voice QOS requirements regarding maximum end-to-end delay and jitter, and the goal of facilitating a lightweight implementation (i.e. low cost) of the Media Adapter. In particular, to satisfy the delay and jitter requirement, any voice compression algorithms (G.729, etc) are implemented at the Media Adapter.

2.2.1 Media Adapter Functions

The Media Adapter performs the following functions:

2.2.1.1 HomePNA Network Interface

The Media Adapter implements the HPNA MAC services described in section 4.15, compliant to the HPNA 2.0 interface specification.

2.2.1.2 Audio Decoding

The Media Adapter receives packetized voice coded according to ITU standards G.711 a-law, G.711 u-law, G.728 or G.729A/B/E and implements the audio decoding function and D/A codec. The specification of vocoder algorithm conveyed in a field of the received packet. Some vocoder algorithms incorporate voice activity detection (VAD) and reduce packet rate accordingly during periods of silence – the audio decoder is responsible for comfort noise generation (CNG) during silence based on spectral characteristics relayed from the encoder.

2.2.1.3 Bridging

The Media Adapter is capable of transferring multiple distinct audio streams between a single handset and the carrier network. The Media Adapter performs audio mixing for delivery to the handset.

2.2.1.4 Jitter Buffer

The Media Adapter performs adaptive jitter buffering for delay equalization in received audio packets. The jitter buffer length need only be sufficient to account for worst case HomePNA network segment jitter; end-to-end carrier network delay equalization is performed by the downstream jitter buffer in the Proxy Gateway.

2.2.1.5 Call Progress Tone Generation

The Media Adapter generates standard call progress tones, including tone cadence, according to state information received from the Proxy Gateway. The list of call progress tones includes:

- Dial tone
- Busy tone
- Ringback tone
- Call Waiting tone
- Stutter Dial Tone
- Reorder (congestion) Tone
- ROH (receiver off hook) Warning Tone
- Confirmation Tone

2.2.1.6 CAS Relay / Generation

The Media Adapter generates Channel Associated Signaling (CAS) to the subscriber line interface circuit (SLIC) according to state information received from the Proxy Gateway. The Media Adapter is capable of generating the following CAS signal states:

- Loop Current Feed
- Loop Current Feed Open
- Power Ringing

- Reverse Loop Current Feed

2.2.1.7 CLASS Signaling

The Media Adapter generates CLASS signals compliant with Bellcore TA-NWT-000030 on to the subscriber line interface according to packetized CLASS signals received from the Proxy Gateway. The control of the timing relationships between CAS signal generation and CLASS signal modulation is the responsibility of the Proxy Gateway. CLASS type 1 (on-hook caller-id message or visual message waiting indicator control) and type 2 (off-hook caller-id message) are supported.

2.2.1.8 Timing Synchronization

The Media Adapter performs two types of timing synchronization during audio encoding, based on synchronization signals relayed from the Proxy Gateway.

1. The 8kHz sample rate of the analog voice codec at the handset is synchronized to a reference clock at the Proxy Gateway. The Proxy Gateway reference clock is synchronized to a network stratum reference clock. This is necessary to eliminate frame slips from clock drift.
2. The generation of encoded voice packets is synchronized to the arrival of the assigned upstream timeslot on the digital carrier network, accounting for any processing delays or jitter introduced by HPNA network access. In the DOCSIS / PacketCable system, this is the arrival of an upstream grant sync for the service flow allocated for the specific voice stream. This is necessary to minimize latency on the upstream path.

2.2.1.9 Audio Encoding

The Media Adapter is responsible for A/D conversion of the analog voice signal and implements the audio encoding function according to ITU standards G.711 a-law, G.711 u-law, G.728 or G.729A/B/E for the generation of packetized voice. The selection of vocoder encoding algorithm is controlled via state information received from the Proxy Gateway and the action of the call discrimination function described below. The minimum frame size (packetization rate) is 10 msec.

2.2.1.10 Echo Cancellation

The Media Adapter must implement line echo cancellation (ECAN) according to ITU standard G.165. The echo cancellation is controlled via state information received from the Proxy Gateway and the local action of the call discrimination function described below.

2.2.1.11 DTMF Detect

The Media Adapter detects DTMF digits and generates DTMF tone on and tone off events to the Proxy Gateway. When the audio encoder is also enabled, DTMF events are passed both in-band as encoded voice, and as out-of-band DTMF state events.

2.2.1.12 Call Discrimination (fax, modem tones)

The Media Adapter performs call discrimination on the ingress analog signal for the detection of fax, modem or TDD tones. The Media Adapter implements ITU standard

V.21, V.25, V.8, and V.18 detectors. The Media Adapter informs the Proxy Gateway of call discrimination events depending if fax modem or TDD tones or human speech are detected; the Proxy Gateway manages the switching to different vocoder algorithms in conjunction with the allocation of appropriate carrier network resources.

2.2.1.13 CAS Relay / Detection

The Media Adapter performs hook-switch monitoring on the local subscriber line interface (SLIC) and relays the state information to the Proxy Gateway. The Media Adapter reports hook-switch state only – the detection of timing of multiple events that represents a hook-flash event is the responsibility of the Proxy Gateway. Debouncing of hook-switch events is performed by the Media Adapter.

2.2.1.14 Performance Statistics

The Media Adapter collects statistics counters on jitter buffer performance and relays this information to the Proxy Gateway at periodic intervals.

2.2.1.15 Capability/Feature Announcement

The Media Adapter informs the Proxy Gateway of supplementary features or capabilities it supports.

2.2.2 Proxy Gateway Functions

The Proxy Gateway performs the following functions:

2.2.2.1 CAS Loop Control

The Proxy Gateway is responsible for timing of state transitions on the Media Adapter loop interface. It is generates ring signal cadence by the timing of ringer on and off events and manages ring-trip removal. It is responsible for managing the timing between CAS state events and CLASS messages for on-hook and off-hook CLASS services, according to Bellcore GR-30. It is responsible for meeting for the ring-trip removal delay requirement.

2.2.2.2 CAS Hook Monitor

The Proxy Gateway performs hook-switch event detection based on the timing of hook-switch events reported from the Media Adapter according to Bellcore GR-506. The Proxy Gateway is able to determine off-hook, on-hook and hook-flash events and report those events to the upstream telephony service. Pulse-dial digit timing is not supported.

2.2.2.3 Downstream Jitter Buffer

The Proxy Gateway performs adaptive jitter buffering for delay equalization in received audio packets. The jitter buffer length must be sufficient to account for worst case end-to-end carrier network jitter.

2.2.2.4 Synchronization Buffer

The Proxy Gateway performs upstream synchronization for packets received from the Media Adapter. Packets received from the Media Adapter are placed in a holding buffer for transmission at the next upstream timeslot/grant interval. The holding buffer accounts for jitter introduced by variable access delays to the HomePNA network. To minimize overall delay and time spent in the synchronization buffer, the Media Adapter arranges to transmit upstream voice packets just in time, making use of the timing synchronization service described in section 5.

2.2.2.5 Network Signaling Protocol

The Proxy Gateway acts as a protocol proxy for the upstream carrier network telephony protocol. The proxy operations will be different depending on the specific upstream telephony protocol in use. In the case of PacketCable 1.0, the Proxy Gateway is responsible for digit collection, reflex operations and MGCP/RTP IP protocol.

2.2.2.5.1 Digit collection

The Proxy Gateway collects single digits from the Media Adapter according to the dial-plan digit map received from the MGC. When the collect digit string matches the digit map, an event is reported to the MGC.

2.2.2.5.2 Reflex operations (MGCP embedded events/signals)

MGCP defines certain "reflex" operations that the MTA initiates without MGC transaction upon detection of specified events e.g. local generation of dial-tone when off-hook is detected. The Proxy Gateway is responsible for initiating reflex operations based on event descriptors received from the MGC.

2.2.2.5.3 MGCP/RTP/IP Protocol

The Proxy Gateway is responsible for protocol proxy and conversion between the carrier network IP-based protocol and the HPNA MAC-based protocol.

2.2.2.5.4 Network Management

The Proxy Gateway is the edge of the service provider's managed network. The Proxy Gateway implements the relevant SNMP MIBs and responds to management operations invoked by the service provider.

3 Model of Operation

The Voice-over-HomeNetwork (VOHN) protocol is conceptually a lightweight Data Link Layer protocol that provides for the reliable transfer of signaling and digital voice payloads. The protocol utilizes the services of the HPNA MAC layer to provide access to the physical media and transparent transfer of link layer frames between a Proxy Gateway and Media Adapters. The frame structure is defined in section 4.1.

The communication path is between a Proxy Gateway and a Media Adapter. There is no provision for direct communication between two Media Adapters (e.g. for home intercom) without relaying through a Proxy Gateway. Except during the discovery and establishment of line bindings, VOHN frames are transported using point-to-point (unicast) MAC station addresses.

The model of operation is similar to Frame Relay Forum agreement FRF-11.1, consisting of the periodic sampling and transfer of the information state of each line termination. The information state is sampled and transferred at a sufficiently high resolution and redundancy to ensure that all state transition events or signals of interest are reliability reported. During static or quiescent periods (no state transitions), the transmitter switches frequency of transmission to a low background rate. During active periods, the transmitter resumes transmission at the higher foreground rate.

Transport of digitized voice is provided with a generalized payload frame format that supports different voice coding algorithms using algorithm-specific "transfer syntax" definitions and procedures. Transfer of supporting information, such as CAS signaling, CLASS messages, dialed digits, call progress tones and performance statistics, is also provided through the use of transfer syntax definitions specific to the information being sent.

The following payload types are used to convey the information state of a line termination:

Type	Meaning
VOICE	Primary encoded voice or data payload
CAS	Channel Associated Signaling – Loop start control
DIGIT	DTMF key down/up detection
CPTONE	Call Progress Tone generation/detection
CLASS	CLASS message relay
MODE	Select and enable encoder algorithm type
STATS	Performance counters
FEATURE	Capability/Feature announcement
FKEY	(future) Function key up/down detection
LED	(future) Local LED on/off control
DISPLAY	(future) Local message display control (e.g. LCD)

4 Definition of Procedures

4.1 Frame Format and Encoding

Signaling and Voice payloads are encoded in frames that are transported as a Link Layer Protocol according to the formats and procedures for HPNA 2.0 Link Layer Framing.

4.1.1 Signaling Frame

All fields are encoded and transported in network byte-order (big-endian). Bit 0 is the least significant bit within a field. Diagrams show MSB bits or octets to the left.

VOHN signaling messages are data link layer frames that are identified by a new IEEE assigned Ethertype value in the frame header.

Field	Length	Meaning
DA	6 octets	Destination Address
SA	6 octets	Source Address
Ethertype	2 octets	*(TBD)* = VOHN Link Control Frame – new IEEE assignment
Type	2 octets	0 = VOHN Signaling Frame
Length	2 octets	Number of additional octets in the signaling frame, starting with Version field and ending with the last octet of the Data Payload field. Minimum is 8.
Version	2 octets	= 0
Line ID	2 octets	Logical line identifier. Identifies a specific line termination.
Timestamp	4 octets	Timestamp. The LSBit of the Timestamp corresponds to a time of 125 us (8 khz).
Payload Element(s)	4-N octets	One or more subfield elements as described below.
PAD	0-36 octets	Padding to make minimum 64 octets HPNA frame length. Any value
FCS	4 octets	Frame check sequence

4.1.2 Payload Element Field Format

Each frame carries one or more payload element fields. Each payload element may be variable length. Multiple payload types may be concatenated within a single frame in any order.

SubField	Length	Meaning
Type	1 octet	General class payload identifier
Subtype	1 octet	Class-specific payload information
Payload Length	2 octets	Number of octets in the Payload field.
Payload	0-N octets	Voice/data payload, depending on type/subtype

4.2 Frame Transmission Procedure

State information is sampled and transferred according to a 10 ms frame clock maintained by the transmitter. The 10 ms frame clock at the Media Adapter is synchronized to the Proxy Gateway upstream timeslot / grant interval through the procedure specified in section 5.

Frames are transmitted at one of two rates:

- Background rate. During static or quiescent periods (no state transitions), the transmitter sends a frame once every 5 seconds.
- Foreground rate. When state information changes, the transmitter sends a frame once every 10 ms. The transmitter remains at foreground rate until a quiescent period of at least 50 ms has elapsed.

The Line-ID field reflects the identifier for the appropriate line termination.

The Timestamp field reflects the incremental time difference between successive frame transmissions.

The number and type of payload subfields conveyed in the frame is described in the transfer syntax for each payload type. Whenever possible, multiple payload elements are concatenated together in a single frame when the transmitter is operating at the foreground rate.

4.3 Voice Payload Transfer Syntax

Voice payload fields transfer packetized voice encoded to ITU standards G.711 a-law, G.711 u-law, G.728 or G.729A/B/E. A single frame contains 10 ms of audio.

4.3.1 Payload format

SubField	Value	Meaning
Type		VOICE
Subtype	1-63	Standard type range: G711 ULAW VAD G711 ALAW VAD G728 G729A G729B SID (G.929B silence identifier) G729E SID (generic VAD silence identifier) G711 ULAW DATA (Voiceband data relay) G711 ALAW DATA (Voiceband data relay) G711 ULAW DATAPREV (previous 10ms payload) G711 ALAW DATAPREV (previous 10ms payload)
	64-95	Test/Experimental type range
	96-128	Vendor-specific format type range
	128-255	Reserved for future use
Payload Length	Variable	based on subtype.
	82	G711 ULAW, ALAW or ULAW DATA payload
	22	G728 payload
	12	G729A payload

	4 tbd 10	G729B SID payload G729E payload SID payload
Payload	variable	Subtype dependent. First two octets are the Call ID, an identifier of the specific call instance for a multi-party bridged call. Default call ID = 0.

4.3.2 Transmission of Voice Payloads

Voice payload fields are transmitted at a 10 ms frame rate while a voice path is established to the line termination. Some vocoder algorithms incorporate voice activity detection (VAD) and reduce packet rate significantly during periods of silence.

The first two octets of the payload field contain the Call ID, an identifier of the specific call instance to allow for multi-party bridged calls at the Media Adapter.

Voice-band data traffic (G711 U/A-LAW DATA) is treated as a special case. Voice-band data is less sensitive to delay, but more sensitive to frame loss than Voice traffic. To increase delivery reliability over the HomePNA segment, frames containing voice-band data contain two payload fields, G711 U/A-LAW DATA containing the voice samples from the current 10 ms period, and G711 U/A-LAW DATAPREV containing a repeat of the voice samples from the immediately previous 10 ms period.

4.3.3 Interpreting Received Voice Payloads

When the receiver gets a VOICE payload, it processes the Voice state based on the timestamp. Possible redundant payload containing voice-band data are identified by a timestamp and payload type and are discarded.

4.4 CAS Transfer Syntax

CAS payload transfers the state of channel associated signaling for standard residential loop start control.

4.4.1 Payload Format

SubField	Value	Meaning
Type		CAS
Subtype	 – 0 4 5 15 5 15	ABCD signaling bit format Ringing RLCF = Reverse Loop Current Feed LCF = Loop Current Feed LCFO = Loop Current Feed Open LO = Loop Open (On-hook) LC = Loop Closed (Off-hook)
Payload Length	0	No additional payload data

The distinction of LO/LCF and LC/LCFO subtypes depends on the direction of transmission.

4.4.2 Transmission of CAS Payloads

CAS Payload field is included in every frame transmission.

4.4.3 Interpreting Received CAS Payloads

When the receiver gets a CAS payload, it processes the CAS state transitions based on the timestamp, e.g. the Proxy Gateway performs hook-flash detection based on the relative timing of CAS state transitions. Pulse-digit dialing is not supported.

4.5 Digit Transfer Syntax

Digit payload transfers the state of the DTMF detector at the Media Adapter. It can also be used to transfer the set the desired state of the DTMF generator at the Media Adapter if required.

4.5.1 Payload Format

SubField	Value	Meaning
Type		DIGIT
Subtype	0	Digit 0 tone on
	1	Digit 1 tone on
	2	Digit 2 tone on
	3	Digit 3 tone on
	4	Digit 4 tone on
	5	Digit 5 tone on
	6	Digit 6 tone on
	7	Digit 7 tone on
	8	Digit 8 tone on
	9	Digit 9 tone on
	10	Digit * tone on
	11	Digit # tone on
	12	Digit A tone on
	13	Digit B tone on
	14	Digit C tone on
	15	Digit D tone on
	255	Tone Off
Payload Length	0	No additional payload data

4.5.2 Transmission of Digit Payloads

The transmitter sends DIGIT payloads to relay DTMF digit signals. When the vocoder function is enabled, DTMF tones will be sent both as DIGIT payloads and encoded in VOICE payloads in the same frame.

In the quiescent ToneOff state, it is not necessary to transmit DIGIT payload at the background frame rate.

4.5.3 Interpreting Received Digit Payloads

A receiver shall interpret the absence of DIGIT payload in a received frame as equivalent to DIGIT ToneOff status.

The Proxy Gateway receiver processes DIGIT payload according to the rules and state of the upstream telephony protocol. In PacketCable-NCS, single digits are detected by examining off and on transitions and processing according to the digit map. In ATT-GR303, DIGIT payload is discarded (digits are relayed as encoded voice).

4.6 Call Progress Tone Transfer Syntax

CP Tone payload is used to set the state of the CP tone generator at the Media Adapter. It is also used to transfer the state of the call discriminator (answer tone, fax tone).

4.6.1 Payload Format

SubField	Value	Meaning
Type		CPTONE
Subtype	0	Dial Tone
	1	Busy Tone
	2	Ringback Tone
	3	Stutter Dial Tone
	4	Message Waiting Tone
	5	Reorder (congestion) Tone
	6	ROH (receiver off hook) Warning Tone
	7	Confirmation Tone
	8	SIT Tone
	9	Calling Card Tone
	100	Answer Tone
	101	Fax Tone
	102	TDD Tone
	255	Tone Off / Idle
Payload Length	0	No additional payload data

4.6.2 Transmission of CP Tone Payloads

The Proxy Gateway transmits CPTONE payload when call progress tones are to be generated by the Media Adapter (e.g. in the case of PacketCable-NCS).

The Media Adapter transmits CPTONE payload to relay call discrimination signals (modem, fax, TDD).

In quiescent ToneOff state, it is not necessary to transmit CPTONE payload at the background frame rate. All other CPTONE subtype are transmitted at the background frame rate.

4.6.3 Interpreting Received CP Tone Payloads

A receiver shall interpret the absence of CPTONE payload in a received frame as equivalent to CPTONE ToneOff status.

4.7 CLASS Message Transfer Syntax

CLASS Message payload is used to relay a CLASS (Caller ID) message to the Media Adapter.

4.7.1 Payload Format

SubField	Value	Meaning
Type		CLASS
Subtype	0 1 2	Class Start Class Body Class End
Payload Length	6-54	Variable length
Payload	variable	Class message contents

4.7.2 Transmission of CLASS Payloads

The Proxy Gateway transmits CLASS payload to transfer a Caller ID message to the Media Adapter.

The Proxy Gateway is responsible for the transfer of frames with relative timing of CAS payload and CLASS payload to conform to Bellcore TA-NWT-000030.

For CLASS I (on-hook callerID), this information is sent between the first and second rings on an analog phone line. For CLASS II (off-hook callerID-on-call-waiting), the CLASS information is sent after the CAS-ACK sequence.

CLASS payload contains 40 ms of CLASS signaling and is sensitive to frame loss. To increase delivery reliability over the HomePNA segment, frames containing CLASS payload are repeated (2 identical copies queued to MAC layer).

4.7.3 Interpreting Received CLASS Payloads

The receiver generates the 1200 bps FSK signal from the payload data immediately on receiving the CLASS Start payload. Possible redundant frames containing CLASS signaling are identified by a duplicate timestamp and are discarded.

4.8 Vocoder Mode Transfer Syntax

The Vocoder Mode provides control over the state and selection of the vocoder algorithm at the Media Adapter.

4.8.1 Payload Format

SubField	Value	Meaning
Type		MODE
Subtype	0 1-63	IDLE/DISABLED Standard type range:

		G711 ULAW VAD G711 ALAW VAD G711 ULAW (VAD disabled) G711 ALAW (VAD disabled) G728 G729A G729B G729E G711 ULAW DATA (Voiceband data relay) G711 ALAW DATA (Voiceband data relay)
	64-95	Test/Experimental type range
	96-128	Vendor-specific format type range
	128-255	Reserved for future use
Payload Length	2	
Payload	2	Call ID

4.8.2 Transmission of Vocoder Mode Payloads

MODE payload is transmitted by the Proxy Gateway to control the voice encoder function at the Media Adapter. The Proxy Gateway must ensure that the Media Adapter is synchronized to the appropriate Proxy Gateway upstream timeslot / grant interval through the procedure specified in section 5 prior to transmitting a non-IDLE MODE state.

The first two octets of the payload field contain the Call ID, an identifier of the specific call instance to allow for multi-party bridged calls at the Media Adapter.

In quiescent IDLE/DISABLED state, it is not necessary to transmit MODE payload at the background frame rate. All other MODE values must continue to be transmitted at the background frame rate.

4.8.3 Interpreting Received Vocoder Mode Payloads

The receiving Media Adapter sets the state of its vocoder as specified in the payload. The Call ID identifies the specific vocoder instance.

The receiving Media Adapter shall interpret the absence of a MODE payload in a received frame as equivalent to MODE IDLE state.

When a non-IDLE to IDLE state transition is detected, the Media Adapter sends a STATS payload 3 times at the foreground frame rate, then clears the performance statistics counters.

4.9 Performance Stats Transfer Syntax

4.9.1 Payload Format

SubField	Value	Meaning
Type		STATS
Subtype	0	TRAFFIC STATS
Payload	14	

Length		
Payload	2 octets 4 octets 4 octets 4 octets	Call ID Packets Sent Packets Received Packets Lost (Packets Expected – Packets Received)

4.9.2 Transmission of Performance Stats Payloads

Performance statistic payload is transmitted periodically by the Media Adapter while in a non-IDLE MODE state. The counters are not reset after transmission. The period is once every (*TBD: 5 seconds?*), regardless of the frame rate background/foreground mode.

The first two octets of the payload field contain the Call ID, an identifier of the specific call instance to allow for multi-party bridged calls at the Media Adapter.

When a non-IDLE to IDLE state transition is detected, the Media Adapter sends a STATS payload 3 times at the foreground frame rate, then clears the performance statistics counters.

4.9.3 Interpreting Received Performance Stats Payloads

4.10 Feature Capability Transfer Syntax

FEATURE payload is transmitted by the Media Adapter to inform the Proxy Gateway about supported features or capabilities.

4.10.1 Payload Format

SubField	Value	Meaning
Type		FEATURE
Subtype	0	
Payload Length	variable	
Payload	2*N	List of payload type/subtype fields supported by the Media Adapter

4.10.2 Transmission of Feature Payloads

The Feature payload is transmitted by the Media Adapter at line binding time (see section 4.12) to inform the Proxy Gateway of supported features and capabilities. It is not necessary to transmit Feature payload at other times.

The payload field contains a list of the type/subtype payloads supported by the Media Adapter. In particular, the list must contain, at a minimum:

- Each VOICE subtype supported (i.e. vocoder types)
- CLASS type, if supported (absent if not supported)
- Each CPTONE subtype supported
- DIGIT subtypes A-D, if supported. (0-9,*,# are assumed base-level features)

- (future) FKEY, LED, DISPLAY subtypes
- Additional feature payload specific subtypes TBD

4.10.3 Interpreting Received Feature Payloads

The receiving Proxy Gateway maintains an information base of feature/capability support per Media Adapter. The Proxy Gateway refers to this information during call establishment procedures with the carrier network.

4.11 Security

Security is not addressed by this protocol. There is no authentication between Proxy Gateway and Media Adapter. There is no privacy encryption of frame payloads.

4.12 Line Binding Procedure

This procedure provides a means to establish a binding between a Proxy Gateway and Media Adapter(s) for each individual line termination. This facilitates the dynamic discovery of the MAC addresses of the Gateway and Media Adapters associated with a particular line termination.

Each line termination is identified by a unique small integer, called the LineID. Each Proxy Gateway or Media Adapter is provisioned with the set of its LineID(s). The provisioning procedure is outside the scope of this specification, but could include e.g. a user-controllable switch, or pre-programmed non-volatile storage, or local "feature code" dial string.

Each network element maintains an information base that binds a Line ID to a MAC Source Address (SA).

At initialization time, or such times when it has no SA value bound against a Line ID, the Media Adapter sends frames at the foreground frame rate and addressed to the broadcast destination address (FF.FF.FF.FF.FF.FF). *(TBD: use a reserved MAC multicast address for Voice/HPNA, instead of broadcast?)*

If a Proxy Gateway receives a frame with DA field = FF.FF.FF.FF.FF.FF and Line ID field belonging the set of Line IDs it serve, it:

1. Creates a local association of the SA field of the received frame with the Line ID.
2. Transmits a response frame with DA = SA.

If the Media Adapter receives a unicast frame with Line ID belonging to the set of Line IDs it serves, it creates a local binding of the SA field of the received frame with the Line ID and restarts its Configuration timer.

If the Media Adapter receives no frame response for foreground frame rate timeout period, it switches to background rate transmission and continues to send frames.

4.13 Failure Detection and Recovery

The Media Adapter maintains a Configuration timer for each Line ID it serves. The Configuration timer is restarted upon reception of a frame with matching Line ID. A suggested value of the Configuration timer is 30 seconds.

Upon expiry of Configuration timer, the Media Adapter clears any SA binding against the Line ID and reinitiates the Line Binding procedure described above. It must also set the physical line interface to an idle condition.

The Proxy Gateway maintains a Configuration timer for each Line ID and SA binding it maintains. The Configuration timer is restarted upon reception of a frame with matching Line ID and SA. A suggested value of the Configuration timer is 30 seconds.

Upon expiry of Configuration timer, the Proxy Gateway clears any SA binding against the Line ID and waits for the Media Adapter to reinitiate the Line Binding procedure described above. It must also set the upstream telephony line termination to an alarm condition.

4.14 Duplicate Line Management Procedure

The Duplicate Line Management procedure is an optional service of the Proxy Gateway that allows multiple Media Adapters to share the same line termination. This allows, for example, incoming calls to ring at more than one handset and be answered at any one, or for an outgoing call on a specific line to originate at different handsets. However, due to limitations imposed by packetized voice encoding delays, it is not possible to perform voice conferencing by mixing/combining audio streams involving more than one Media Adapter.

A Proxy Gateway which supports this service maintains multiple bindings for each line termination that it serves.

4.14.1 Downstream CAS signaling

The Proxy Gateway replicates and transmits CAS and CLASS payload frames to each Media Adapter DA which is bound to the Line ID. In this way, ringing and caller ID messages are distributed to each Media Adapter bound to the line.

4.14.2 Off-hook Seizure

Any Media Adapter bound to the line may transmit a CAS off-hook payload and exclusively seize the line. Subsequent Media Adapters that may go off-hook may be handled by the Proxy Gateway according to different policies, e.g.:

- Most recent off-hook event seizes the line. This allows for informal call transfer between Media Adapters on the HPNA network.
- Proxy Gateway maintains a LIFO stack of off-hook events and transfers the call between Media Adapters according to top-of-stack position.
- Proxy Gateway implements in-home conference bridging by transcoding and merging voice streams, at the expense of additional delay.

The Proxy Gateway transmits payload frames (other than CAS or CLASS) and accepts received payload frames only with the seized Media Adapter. A network warning tone CPTONE payload should be sent to off-hook, unseized Media Adapters. When a CAS on-hook payload is received, the Media Adapter releases the seizure of the line.

4.15 MAC Layer Service Access

The Voice over HomePNA protocol utilizes the services of the HPNA MAC layer to provide access to the physical media and transparent transfer of link layer frames. The MAC layer provides the following services:

1. Point-to-Point Information Transfer

   Frames are transferred between Proxy Gateway and Media Adapter using point-to-point unicast addressing. The DA field value can be specified in transmit frame requests. The SA field value can be retrieved from receive frame indications.

2. Broadcast Data Transfer

   Line Binding procedure requires that frames be transmitted using the broadcast addressing. The broadcast address can be specified in transmit frame requests. An indication of broadcast/unicast addressing is provided in receive frame indications.

3. Frame Error Detection

   The MAC layer performs error detection on received frames and silently discards frames with errors.

4. Quality of Service

   The MAC layer provides differentiated service levels that meet a tightly-bounded latency requirement. Voice frames are transmitted using the highest priority level PRI=7, guaranteeing access to the media ahead of all lower priority traffic. Latency requirements are met assuming exclusive use of this level for voice and a-priori knowledge of the aggregate bandwidth requirements per call.

5 Time Synchronization

Signaling frames and procedures are defined to permit time synchronization between the Proxy Gateway and Media Adapter. The time synchronization procedures enable two types of time synchronization:

1. The 8kHz sample rate of the analog voice codec at the handset is synchronized to a reference clock at the Proxy Gateway.
2. The generation of encoded voice packets at the Media Adapter is synchronized to the arrival of the assigned upstream timeslot at the Proxy Gateway from the digital carrier network, accounting for any processing delays or jitter introduced by HPNA network access. In the DOCSIS / PacketCable system, this is the arrival of an upstream grant sync for the service flow allocated for the specific voice stream.

5.1 Overview of Codec Clock Synchronization

The Proxy Gateway implements a counter/timer that is sync-locked to the network stratum reference source. The HPNA MAC transmitter in the Proxy Gateway implements a function to read and latch the value of the counter/timer into a Master Timestamp Register at the exact time of transmission of a frame marked with the "Latch Timestamp" (LTS) descriptor bit.

The Media Adapter implements a counter/timer which is subdivided to derive the Codec clock. The HPNA MAC receiver in the Media Adapter implements a function to read and latch the value of the counter/timer into a Receive Timestamp Register upon the receipt of a frame. The Receive Timestamp Register is logically part of the receive status word of each received frame.

The timing information is conveyed to the Media Adapter via a pair of messages. The Proxy Gateway periodically transmits a Timestamp Sync (TSM) frame with the LTS descriptor set, then reads and transmits the latched Master Timestamp register value in a subsequent Timestamp Report (TRM) frame.

The Media Adapter reads and saves the Receive Timestamp register values of Timestamp Sync frames, and builds a database of corresponding Receive and Master timestamp pairs from the received TSM and TRM frames.

The Media Adapter periodically calculates:

$$\text{frequency error} = [(R_2 - R_1) / (M_2 - M_1)] - 1$$

The method by which the frequency error adjustment is then applied to the Media Adapter local codec clock is implementation-dependent.

5.2 Overview of Packet Timeslot Grant Synchronization

The Proxy Gateway implements a function to read and latch the value of the reference counter/timer into a Grant Timestamp register upon the occurrence of a selected timeslot

grant sync signal from the upstream network (i.e. SID match and Grant sync). The Proxy Gateway is aware of the mapping of upstream timeslot grant to specific Media Adapter and line ID.

The Media Adapter implements a timer that generates a local frame sync signal at the expected voice frame rate. This timer is derived from the local codec clock.

The relative timing of the upstream grant sync signal is conveyed to the Media Adapter prior to enabling the voice encoder, but after the establishment of the upstream service flow. The timing offset is adjusted to account for internal processing cycles needed each by the Proxy Gateway and the Media Adapter, and allowing for worst case voice frame latency on the HPNA media.

When the Proxy Gateway needs to send the timeslot grant sync timing information, it will latch the grant timestamp value and adjust the value to account for worst case HPNA media latency and the internal processing time to receive and forward voice frames to the upstream network interface The adjusted grant timestamp is transmitted to the Media Adapter in a Timestamp Report (TRM) frame.

The Media Adapter calculates an absolute time offset from the difference in the Receive and Master timestamps, and calculates a future local frame sync time as:

Frame Sync = Grant timestamp + offset + voice frame period - processing time.

The method by which the Frame Sync adjustment is then applied to the Media Adapter voice encoder is implementation-dependent.

5.3 Timestamp Sync Frame Format

Field	Length	Meaning
DA	6 octets	Destination Address
SA	6 octets	Source Address
Ethertype	2 octets	*(TBD)* = VOHN Link Control Frame – new IEEE assignment
Type	2 octets	1 = Timestamp Sync Message
Length	2 octets	= 4
Version	2 octets	= 0
SeqNum	2 octets	Timestamp Sync Message Sequence Number
Pad		Any value octet
FCS	4 octets	Frame Check Sequence

5.4 Timestamp Report Frame Format

Field	Length	Meaning
DA	6 octets	Destination Address
SA	6 octets	Source Address
Ethertype	2 octets	(*TBD*) = VOHN Link Control Frame – new IEEE assignment
Type	2 octets	2 = Timestamp Report Message
Length	2 octets	Number of additional octets in the signaling frame, starting with Version field and ending with the last octet of the Data Payload field. Minimum is 2.
Version	2 octets	= 0
TSMSeqNum	2 octets	Sequence number of TSM to which the Timestamp in this message is applicable.
Timestamp	4 octets	Timestamp of a previously transmitted Timestamp Report Message, corresponding to TSMSeqNum.
Frequency	2 octets	Resolution of the timestamp and Gtimestamp fields, in ticks/1.000 ms. For example, value 32768 corresponds to one clock tick at 32.768 Mhz, in which the LSBit of the Timestamp corresponds to a time of 0.030517578125μsec. The Timestamp will rollover every 131 seconds = 2.2 minutes
NumGrants	2 octets	Number of Grant Timestamps specified in the payload of this control message. NumGrants may be zero. Each grant timestamp is accompanied by a Line ID and Call ID field. Including the Grant Timestamp, the total for each grant timestamp is 8 bytes.
Line ID	2 octets	Identifier of the Line termination associated with the immediately following GTimestamp.
Call ID	2 octets	Identifier of the call instance on the Line termination associated with the immediately following GTimestamp.
GrantTimestamp	4 octets	Grant Timestamp corresponding to the immediately preceding Line ID. This is the time at which the Proxy Gateway wishes to receive a future constant bit rate service flow packet in order to minimize delivery latency to subsequent delivery to a synchronous network. The time value corresponds to the time at the timing master. Additional packets for the identified service flow are expected to arrive at periodic intervals measured from this time.
...		{additional instances of {LineID, Call ID, Grant Timestamp} field tuples]
Pad		Any value octet
FCS	4 octets	Frame Check Sequence

5.5 Transmission of Timestamp Frames

The Proxy Gateway transmits time synchronization frames (Timestamp Sync Message and Timestamp Report Message) on a periodic rate continuously.

Frames are transmitted to the broadcast MAC address using MAC priority level 6. (*TBD: use a reserved MAC multicast address instead of broadcast?)*

5.5.1 Clock Synchronization

Time sync messages are always transmitted in pairs, according to the following procedure:

The Proxy Gateway maintains a Time Sync timer and a sequence number counter, SeqNum. Upon expiry of the time sync timer, the Proxy Gateway:

- restarts the Time Sync timer with period 1 second,
- increments SeqNum = SeqNum + 1,
- formats a Timestamp Sync Message frame with the current value of SeqNum,
- marks the frame with the LTS = 1 descriptor and
- transmits the TSM frame.

The Proxy Gateway then:

- reads the value of the Master Timestamp register,
- formats a Timestamp Report Message frame with the current values of SeqNum and Master Timestamp, and
- transmits the TRM frame.

5.5.2 Timeslot Grant Synchronization

Upon the establishment or re-establishment of an upstream service flow for a media stream, the Proxy Gateway:

- obtains the grant timestamp for the service flow from the Grant Timestamp register,
- adjusts the grant timestamp by a known constant equal to the internal processing time to receive and forward an upstream voice packet,
- adjusts the grant timestamp by a known constant equal to the worst case HPNA media transmission delay,
- formats a Timestamp Report Message frame as above, including the additional Grant Timestamp and associated Line ID and Call ID fields, and
- transmits 3 copies the TRM frame.

TRM frames containing a Grant Timestamp are transmitted immediately (without waiting for the Time Sync timer to expire).

5.6 Reception of Timestamp Frames

A Media Adapter derives clock and grant timing information from received Timestamp Sync and Timestamp Report message frames. Frames which are received with an MAC source address (SA field) that do not match the expected Proxy Gateway are discarded.

5.6.1 Clock Synchronization

The Media Adapter maintains an information base of {SeqNum, Receive timestamp, Master timestamp} tuples. The most recent 2 tuples are retained; older tuples are discarded.

Upon receipt of a Timestamp Sync Message frame, the Media Adapter reads the Receive Timestamp receive status word, and enters the {SeqNum, Receive Timestamp} tuple into its information base.

Upon receipt of a Timestamp Report Message frame, the Media Adapter:

- locates the tuple associated with the received sequence number, SeqNum, from its information base,
- enters the Master timestamp value in the corresponding tuple in the information base
- calculates a codec clock frequency error:
  $\text{frequency error} = [(R_{seqnum} - R_{(seqnum-1)}) / (M_{seqnum} - M_{(seqnum-1)})] - 1$
- adjusts the local clock frequency as necessary (implementation-dependent)

5.6.2 Timeslot Grant Synchronization

When the Media Adapter receives a Timestamp Report Message frame containing a Grant Timestamp, the Media Adapter

- examines the SeqNum field and discards the message if a duplicate received frame and takes no further action
- examines the Line ID and Call ID field and discards the message if no match to an existing voice call
- calculates the time delta to the next local frame sync signal as follows:

  $\text{Frame sync time} = \text{Grant Timestamp} + T_{offset} + VF - K_{CPU}$

  where

  T_{offset} = Receive Timestamp – Master Timestamp (absolute time offset)

  K_{CPU} = a known constant equal to the Media Adapter internal processing time to prepare an upstream voice packet

  $\overline{VF}$ = voice frame period
- adjusts the local frame sync timing as necessary (implementation-dependent)

6 Example Call Flows

This section gives some call flow examples as an aid to understanding the protocol. A sketch of frame payload contents is shown, but total frames transmitted due to foreground/background frame rate changes is not shown.

6.1 PacketCable-NCS Network

The following examples assume an upstream network conforming to PacketCable-NCS/MGCP signaling.

6.1.1 Outgoing Call Origination

Call State	Proxy Gateway		Media Adapter
Idle	CAS: LCF	→	
		←	CAS: LO (onhook)
Offhook		←	CAS: LC (offhook)
Dialtone	CAS: LCF CPTONE: Dialtone	→	
Dialing		←	CAS: LC DIGIT: on
	CAS: LCF CPTONE: off	→	
		←	CAS: LC DIGIT: off
		←	CAS: LC DIGIT: on
		←	CAS: LC DIGIT: off
Waiting	CAS: LCF CPTONE: ringback	→	
Remote Answer	CAS: LCF CPTONE: off MODE: G729A	→	
Talk	CAS: LCF MODE: G729A VOICE:G729A	←→	CAS: LC VOICE: G729A

6.1.2 Incoming Call Answer

Call State	Proxy Gateway		Media Adapter
Idle	CAS: LCF	→	
		←	CAS: LO (onhook)
Alert	CAS: Ring	→	
	CAS: LCF	→	
Caller ID	CAS: LCF CLASS: msg	→	
	CAS: Ring	→	
	CAS: LCF	→	
Answer		←	CAS: LC (offhook)

Talk	CAS: LCF MODE:G729A	→	
	CAS: LCF MODE: G729A VOICE:G729A	←→	CAS: LC VOICE: G729A

6.1.3 Call Termination

Call State	Proxy Gateway		Media Adapter
Talk	CAS: LCF MODE: G729A VOICE:G729A	←→	CAS: LC VOICE: G729A
Hangup		←	CAS: LO (onhook) VOICE: G729A
	CAS: LCF MODE:IDLE	→	
Idle	CAS: LCF	→	
		←	CAS: LO (onhook)

6.1.4 Feature Activation e.g. Call Waiting and Transfer

Call State	Proxy Gateway		Media Adapter
Talk	CAS: LCF MODE: G729A VOICE:G729A	←→	CAS: LC VOICE: G729A
Call Waiting Indicator	CAS: LCF MODE: G729A CPTONE: call wait tone	←→	CAS: LC VOICE: G729A
	CAS: LCF MODE: G729A CPTONE: off	←→	CAS: LC VOICE: G729A
	CAS: LCF MODE: G729A VOICE:G729A	←→	CAS: LO (onhook) VOICE: G729A
Flash detect	CAS: LCF MODE: G729A VOICE:G729A	←→	CAS: LC (offhook) VOICE: G729A
Talk	CAS: LCF MODE: G729A VOICE:G729A	←→	CAS: LC VOICE: G729A

6.2 Cable GR-303 Network

The following examples assume an upstream network using GR-303 signaling.

6.2.1 Outgoing Call Origination

Call State	Proxy Gateway		Media Adapter
Idle	CAS: LCF	→	
		←	CAS: LO (onhook)
Offhook		←	CAS: LC (offhook)
Dialtone (recv as G711U payload from network)	CAS: LCF MODE: G711U VOICE:G711U	→	
Dialing (DIGIT evts discarded; Tones sent as G711U payload)	CAS: LCF MODE: G711U VOICE:G711U	←→	CAS: LC DIGIT: on VOICE: G711U
	CAS: LCF MODE: G711U VOICE:G711U	←→	CAS: LC DIGIT: off VOICE: G711U
	CAS: LCF MODE: G711U VOICE:G711U	←→	CAS: LC DIGIT: on VOICE: G711U
	CAS: LCF MODE: G711U VOICE:G711U	←→	CAS: LC DIGIT: off VOICE: G711U
Waiting (Ringback tone recv as G711U payload from network)	CAS: LCF MODE: G711U VOICE:G711U	←→	CAS: LC VOICE:G711U
Remote Answer (Cut-through by network)	CAS: LCF MODE: G711U VOICE:G711U	←→	CAS: LC VOICE:G711U
Talk	CAS: LCF MODE: G711U VOICE:G711U	←→	CAS: LC VOICE: G711U

6.2.2 Incoming Call Answer

Call State	Proxy Gateway		Media Adapter
Idle	CAS: LCF	→	
		←	CAS: LO (onhook)
Alert	CAS: Ring MODE: G711U VOICE:G711U	→	
(ring-off delay before caller id)	CAS: LCF MODE: G711U VOICE:G711U	←→	CAS: LO VOICE:G711U
Caller ID (encoded in G711U payload from network)	CAS: LCF MODE: G711U VOICE:G711U	←→	CAS: LO VOICE:G711U
	CAS: Ring MODE: G711U	←→	CAS: LO VOICE:G711U

	VOICE:G711U		
	CAS: LCF MODE: G711U VOICE:G711U	←→	CAS: LO VOICE:G711U
Answer (Cut-through by network)	CAS: LCF MODE: G711U VOICE:G711U	←→	CAS: LC (offhook) VOICE:G711U
Talk	CAS: LCF MODE: G711U VOICE:G711U	←→	CAS: LC VOICE:G711U

6.2.3 Call Termination

Call State	Proxy Gateway		Media Adapter
Talk	CAS: LCF MODE: G711U VOICE:G711U	←→	CAS: LC VOICE: G711U
Hangup		←	CAS: LO (onhook) VOICE: G711U
	CAS: LCF MODE:IDLE	→	
Idle	CAS: LCF	→	
		←	CAS: LO (onhook)

6.2.4 Feature Activation e.g. Call Waiting and Transfer

Call State	Proxy Gateway		Media Adapter
Talk	CAS: LCF MODE: G711U VOICE:G711U	←→	CAS: LC VOICE: G711U
Call Waiting Indicator (tone recv as G711U payload from network)	CAS: LCF MODE: G711U VOICE: G711U	←→	CAS: LC VOICE: G711U
	CAS: LCF MODE: G711U VOICE: G711U	←→	CAS: LC VOICE: G711U
	CAS: LCF MODE: G711U VOICE: G711U	←→	CAS: LO (onhook) VOICE: G711U
Flash detect	CAS: LCF MODE: G711U VOICE: G711U	←→	CAS: LC (offhook) VOICE: G711U
Talk	CAS: LCF MODE: G711U VOICE: G711U	←→	CAS: LC VOICE: G711U

7 Outstanding Issues

1. Use of HPNA priorities.

   Not all frames need to be transmitted at highest priority. E.g. Signaling frames without voice payload can probably be sent at lower priority. Need to examine all interactions. The main concern is additional priority 7 traffic that may overload the delay budget when >4 Media Adapters are present, and all cause signaling events (e.g. 4 calls in progress, and 5^{th} adapter goes off-hook)

2. Use of multicast addresses.

   It is possible to eliminate use of broadcast frames by using one or more multicast address(es) instead. This would require well-known addresses to be allocated and reserved for VoHN. At a minimum, use 2 multicast addresses: 1) Proxy Gateway registration address, 2) TimeStamp messages sent to Media Adapters.

3. Party-line conferencing, call transfer

   A common expected scenario is the ad-hoc party-line and transfer of calls between multiple handsets on the same line. One method to implement this would require merging of voice streams using DSP resources at the Proxy Gateway, but doesn't meet the delay budget constraint.

4. Addition of maintenance/test features – e.g. loop continuity test

   During reviews with MSO technical advisors, it was suggested that loop test and other maintenance features should be provided.

37367/CAG/B600

2.3.1. An Embodiment of a Signal Processing System

The exemplary signal processing system can be implemented with a programmable DSP software architecture as shown in FIG. 7. This architecture has a DSP 17 with memory 18 at the core, a number of network channel interfaces 19 and telephony interfaces 20, and a host 21 that may reside in the DSP itself or on a separate microcontroller. The network channel interfaces 19 provide multi-channel access to the packet based network. The telephony interfaces 23 can be connected to a circuit-switched network interface such as a PSTN system, or directly to any telephony device. The programmable DSP is effectively hidden within the embedded communications software layer. The software layer binds all core DSP algorithms together, interfaces the DSP hardware to the host, and provides low level services such as the allocation of resources to allow higher level software programs to run.

An exemplary multi-layer software architecture operating on a DSP platform is shown in FIG.8. A user application layer 26 provides overall executive control and system management, and directly interfaces a DSP server 25 to the host 21 (see to FIG. 7). The DSP server 25 provides DSP resource management and telecommunications signal processing. Operating below the DSP server layer are a number of physical devices (PXD) 30a, 30b, 30c. Each PXD provides an interface between the DSP server 25 and an external telephony device (not shown) via a hardware abstraction layer (HAL) 34.

The DSP server 25 includes a resource manager 24 which receives commands from, forwards events to, and exchanges data with the user application layer 26. The user application layer 26 can either be resident on the DSP 17 or alternatively on the host 21 (see FIG. 7), such as a microcontroller. An application programming interface 27 (API) provides a software interface between the user application layer 26 and the resource manager 24. The resource manager 24 manages the internal / external program and data memory of the DSP 17. In addition the resource manager dynamically allocates DSP resources, performs command routing as well as other general purpose functions.

The DSP server 25 also includes virtual device drivers (VHDs) 22a, 22b, 22c. The VHDs are a collection of software objects that control the operation of and provide the facility for real time signal processing. Each VHD 22a, 22b, 22c includes an inbound and outbound media queue (not shown) and a library of signal processing services specific to that VHD 22a, 22b, 22c. In the described exemplary embodiment, each VHD 22a, 22b, 22c is a complete self-contained

37367/CAG/B600

software module for processing a single channel with a number of different telephony devices. Multiple channel capability can be achieved by adding VHDs to the DSP server 25. The resource manager 24 dynamically controls the creation and deletion of VHDs and services.

A switchboard 32 in the DSP server 25 dynamically inter-connects the PXDs 30a, 30b, 30c with the VHDs 22a, 22b, 22c . Each PXD 30a, 30b, 30c is a collection of software objects which provide signal conditioning for one external telephony device. For example, a PXD may provide volume and gain control for signals from a telephony device prior to communication with the switchboard 32. Multiple telephony functionalities can be supported on a single channel by connecting multiple PXDs, one for each telephony device, to a single VHD via the switchboard 32. Connections within the switchboard 32 are managed by the user application layer 26 via a set of API commands to the resource manager 24. The number of PXDs and VHDs is expandable, and limited only by the memory size and the MIPS (millions instructions per second) of the underlying hardware.

A hardware abstraction layer (HAL) 34 interfaces directly with the underlying DSP 17 hardware (see FIG. 7) and exchanges telephony signals between the external telephony devices and the PXDs. The HAL 34 includes basic hardware interface routines, including DSP initialization, target hardware control, codec sampling, and hardware control interface routines. The DSP initialization routine is invoked by the user application layer 26 to initiate the initialization of the signal processing system. The DSP initialization sets up the internal registers of the signal processing system for memory organization, interrupt handling, timer initialization, and DSP configuration. Target hardware initialization involves the initialization of all hardware devices and circuits external to the signal processing system. The HAL 34 is a physical firmware layer that isolates the communications software from the underlying hardware. This methodology allows the communications software to be ported to various hardware platforms by porting only the affected portions of the HAL 34 to the target hardware.

The exemplary software architecture described above can be integrated into numerous telecommunications products. In an exemplary embodiment, the software architecture is designed to support telephony signals between telephony devices (and/or circuit switched networks) and packet based networks. A network VHD (NetVHD) is used to provide a single channel of operation and provide the signal processing services for transparently managing voice,

37367/CAG/B600

fax, and modem data across a variety of packet based networks. More particularly, the NetVHD encodes and packetizes DTMF, voice, fax, and modem data received from various telephony devices and/or circuit switched networks and transmits the packets to the user application layer. In addition, the NetVHD disassembles DTMF, voice, fax, and modem data from the user application layer, decodes the packets into signals, and transmits the signals to the circuit switched network or device.

An exemplary embodiment of the NetVHD operating in the described software architecture is shown in FIG. 9. The NetVHD includes four operational modes, namely voice mode 36, voiceband data mode 37, fax relay mode 40, and data relay mode 42. In each operational mode, the resource manager invokes various services. For example, in the voice mode 36, the resource manager invokes call discrimination 44, packet voice exchange 48, and packet tone exchange 50. The packet voice exchange 48 may employ numerous voice compression algorithms, including, among others, Linear 128 kbps, G.711 u-law/A-law 64 kbps (ITU Recommendation G.711 (1988) - Pulse code modulation (PCM) of voice frequencies), G.726 16/24/32/40 kbps (ITU Recommendation G.726 (12/90) - 40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)), G.729A 8 kbps (Annex A (11/96) to ITU Recommendation G.729 - Coding of speech at 8 kbit/s using conjugate structure algebraic-code-excited linear-prediction (CS-ACELP) – Annex A: Reduced complexity 8 kbit/s CS-ACELP speech codec), and G.723 5.3/6.3 kbps (ITU Recommendation G.723.1 (03/96) - Dual rate coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s). The contents of each of the foregoing ITU Recommendations being incorporated herein by reference as if set forth in full.

The packet voice exchange 48 is common to both the voice mode 36 and the voiceband data mode 37. In the voiceband data mode 37, the resource manager invokes the packet voice exchange 48 for exchanging transparently data without modification (other than packetization) between the telephony device (or circuit switched network) and the packet based network. This is typically used for the exchange of fax and modem data when bandwidth concerns are minimal as an alternative to demodulation and remodulation. During the voiceband data mode 37, the human speech detector service 59 is also invoked by the resource manager. The human speech detector 59 monitors the signal from the near end telephony device for speech. In the event that speech is detected by the human speech detector 59, an event is forwarded to the resource manager which, in turn, causes the resource manager to terminate the human speech detector service 59 and invoke the appropriate services for the voice mode 36 (i.e., the call discriminator, the packet tone exchange, and the packet voice exchange).

37367/CAG/B600

In the fax relay mode 40, the resource manager invokes a fax exchange 52 service. The packet fax exchange 52 may employ various data pumps including, among others, V.17 which can operate up to 14,400 bits per second, V.29 which uses a 1700-Hz carrier that is varied in both phase and amplitude, resulting in 16 combinations of 8 phases and 4 amplitudes which can operate up to 9600 bits per second, and V.27ter which can operate up to 4800 bits per second. Likewise, the resource manager invokes a packet data exchange 54 service in the data relay mode 42. The packet data exchange 52 may employ various data pumps including, among others, V.22bis/V.22 with data rates up to 2400 bits per second, V.32bis/V.32 which enables full-duplex transmission at 14,400 bits per second, and V.34 which operates up to 33,600 bits per second. The ITU Recommendations setting forth the standards for the foregoing data pumps are incorporated herein by reference as if set forth in full.

In the described exemplary embodiment, the user application layer does not need to manage any service directly. The user application layer manages the session using high-level commands directed to the NetVHD, which in turn directly runs the services. However, the user application layer can access more detailed parameters of any service if necessary to change, by way of example, default functions for any particular application.

In operation, the user application layer opens the NetVHD and connects it to the appropriate PXD. The user application then may configure various operational parameters of the NetVHD, including, among others, default voice compression (Linear, G.711, G.726, G.723.1, G.723.1A, G.729A, G.729B), fax data pump (Binary, V.17, V.29, V.27ter), and modem data pump (Binary, V.22bis, V.32bis, V.34). The user application layer then loads an appropriate signaling service (not shown) into the NetVHD, configures it and sets the NetVHD to the On-hook state.

In response to events from the signaling service (not shown) via a near end telephony device (hookswitch), or signal packets from the far end, the user application will set the NetVHD to the appropriate off-hook state, typically voice mode. In an exemplary embodiment, if the signaling service event is triggered by the near end telephony device, the packet tone exchange will generate dial tone. Once a DTMF tone is detected, the dial tone is terminated. The DTMF tones are packetized and forwarded to the user application layer for transmission on the packet based network. The packet tone exchange could also play ringing tone back to the near end telephony device (when a far end telephony device is being rung), and a busy tone if the far end

telephony device is unavailable. Other tones may also be supported to indicate all circuits are busy, or an invalid sequence of DTMF digits were entered on the near end telephony device.

Once a connection is made between the near end and far end telephony devices, the call discriminator is responsible for differentiating between a voice and machine call by detecting the presence of a 2100 Hz. tone (as in the case when the telephony device is a fax or a modem), a 1100 Hz. tone or V.21 modulated high level data link control (HDLC) flags (as in the case when the telephony device is a fax). If a 1100 Hz. tone, or V.21 modulated HDLC flags are detected, a calling fax machine is recognized. The NetVHD then terminates the voice mode 36 and invokes the packet fax exchange to process the call. If however, 2100 Hz tone is detected, the NetVHD terminates voice mode and invokes the packet data exchange.

The packet data exchange service further differentiates between a fax and modem by continuing to monitor the incoming signal for V.21 modulated HDLC flags, which if present, indicate that a fax connection is in progress. If HDLC flags are detected, the NetVHD terminates packet data exchange service and initiates packet fax exchange service. Otherwise, the packet data exchange service remains operative. In the absence of an 1100 or 2100 Hz. tone, or V.21 modulated HDLC flags the voice mode remains operative.

A. The Voice Mode

The services invoked by the network VHD in the voice mode and the associated PXD is shown schematically in FIG. 10. In the described exemplary embodiment, the PXD 60 provides two way communication with a telephone or a circuit switched network, such as a PSTN line (e.g. DS0) carrying a 64kb/s pulse code modulated (PCM) signal, i.e., digital voice samples.

The incoming PCM signal 60a is initially processed by the PXD 60 to remove far end echos. As the name implies, echos in telephone systems is the return of the talker's voice resulting from the operation of the hybrid with its two-four wire conversion. If there is low end-to-end delay, echo from the far end is equivalent to side-tone (echo from the near-end), and therefore, not a problem. Side-tone gives users feedback as to how loud they are talking, and indeed, without side-tone, users tend to talk too loud. However, far end echo delays of more than about 10 to 30 msec significantly degrade the voice quality and are a major annoyance to the user.

37367/CAG/B600

An echo canceller 70 is used to remove echos from far end speech present on the incoming PCM signal 60a before routing the incoming PCM signal 60a back to the far end user. The echo canceller 70 samples an outgoing PCM signal 60b from the far end user, filters it, and combines it with the incoming PCM signal 60a. Preferably, the echo canceller 70 is followed by a non-linear processor (NLP) 72 which may mute the digital voice samples when far end speech is detected in the absence of near end speech. The echo canceller 70 may also inject comfort noise which in the absence of near end speech may be roughly at the same level as the true background noise or at a fixed level.

After echo cancellation, the power level of the digital voice samples is normalized by an automatic gain control (AGC) 74 to ensure that the conversation is of an acceptable loudness. Alternatively, the AGC can be performed before the echo canceller 70, however, this approach would entail a more complex design because the gain would also have to be applied to the sampled outgoing PCM signal 60b. In the described exemplary embodiment, the AGC 74 is designed to adapt slowly, although it should adapt fairly quickly if overflow or clipping is detected. The AGC adaptation should be held fixed if the NLP 72 is activated.

After AGC , the digital voice samples are placed in the media queue 66 in the network VHD 62 via the switchboard 32'. In the voice mode, the network VHD 62 invokes three services, namely call discrimination, packet voice exchange, and packet tone exchange. The call discriminator 68 analyzes the digital voice samples from the media queue to determine whether a 2100 Hz, a 1100 Hz. tone or V.21 modulated HDLC flags are present. As described above with reference to FIG. 9, if either tone or HDLC flags are detected, the voice mode services are terminated and the appropriate service for fax or modem operation is initiated. In the absence of a 2100 Hz, a 1100 Hz. tone, or HDLC flags, the digital voice samples are coupled to the encoder system which includes a voice encoder 82, a voice activity detector (VAD) 80, a comfort noise estimator 81, a DTMF detector 76, a call progress tone detector 77 and a packetization engine 78.

Typical telephone conversations have as much as sixty percent silence or inactive content. Therefore, high bandwidth gains can be realized if digital voice samples are suppressed during these periods. A VAD 80, operating under the packet voice exchange, is used to accomplish this function. The VAD 80 attempts to detect digital voice samples that do not contain active speech. During periods of inactive speech, the comfort noise estimator 81 couples silence identifier (SID)

37367/CAG/B600

packets to a packetization engine 78. The SID packets contain voice parameters that allow the reconstruction of the background noise at the far end.

From a system point of view, the VAD 80 may be sensitive to the change in the NLP 72. For example, when the NLP 72 is activated, the VAD 80 may immediately declare that voice is inactive. In that instance, the VAD 80 may have problems tracking the true background noise level. If the echo canceller 70 generates comfort noise during periods of inactive speech, it may have a different spectral characteristic from the true background noise. The VAD 80 may detect a change in noise character when the NLP 72 is activated (or deactivated) and declare the comfort noise as active speech. For these reasons, the VAD 80 should be disabled when the NLP 72 is activated. This is accomplished by a "NLP on" message 72a passed from the NLP 72 to the VAD 80.

The voice encoder 82, operating under the packet voice exchange, can be a straight 16 bit PCM encoder or any voice encoder which supports one or more of the standards promulgated by ITU. The encoded digital voice samples are formatted into a voice packet (or packets) by the packetization engine 78. These voice packets are formatted according to an applications protocol and outputted to the host (not shown). The voice encoder 82 is invoked only when digital voice samples with speech are detected by the VAD 80. Since the packetization interval may be a multiple of an encoding interval, both the VAD 80 and the packetization engine 78 should cooperate to decide whether or not the voice encoder 82 is invoked. For example, if the packetization interval is 10 msec and the encoder interval is 5 msec (a frame of digital voice samples is 5 ms), then a frame containing active speech should cause the subsequent frame to be placed in the 10 ms packet regardless of the VAD state during that subsequent frame. This interaction can be accomplished by the VAD 80 passing an "active" flag 80a to the packetization engine 78, and the packetization engine 78 controlling whether or not the voice encoder 82 is invoked.

In the described exemplary embodiment, the VAD 80 is applied after the AGC 74. This approach provides optimal flexibility because both the VAD 80 and the voice encoder 82 are integrated into some speech compression schemes such as those promulgated in ITU Recommendations G.729 with Annex B VAD (March 1996) - Coding of Speech at 8 kbits/s Using Conjugate-Structure Algebraic-Code-Exited Linear Prediction (CS-ACELP), and G.723.1 with Annex A VAD (March 1996) - Dual Rate Coder for Multimedia Communications

Transmitting at 5.3 and 6.3 kbit/s, the contents of which is hereby incorporated by reference as through set forth in full herein.

Operating under the packet tone exchange, a DTMF detector 76 determines whether or not there is a DTMF signal present at the near end. The DTMF detector 76 also provides a pre-detection flag 76a which indicates whether or not it is likely that the digital voice sample might be a portion of a DTMF signal. If so, the pre-detection flag 76a is relayed to the packetization engine 78 instructing it to begin holding voice packets. If the DTMF detector 76 ultimately detects a DTMF signal, the voice packets are discarded, and the DTMF signal is coupled to the packetization engine 78. Otherwise the voice packets are ultimately released from the packetization engine 78 to the host (not shown). The benefit of this method is that there is only a temporary impact on voice packet delay when a DTMF signal is pre-detected in error, and not a constant buffering delay. Whether voice packets are held while the pre-detection flag 76a is active could be adaptively controlled by the user application layer.

Similarly, a call progress tone detector 77 also operates under the packet tone exchange to determine whether a precise signaling tone is present at the near end. Call progress tones are those which indicate what is happening to dialed phone calls. Conditions like busy line, ringing called party, bad number, and others each have distinctive tone frequencies and cadences assigned them. The call progress tone detector 77 monitors the call progress state, and forwards a call progress tone signal to the packetization engine to be packetized and transmitted across the packet based network. The call progress tone detector may also provide information regarding the near end hook status which is relevant to the signal processing tasks. If the hook status is on hook, the VAD should preferably mark all frames as inactive, DTMF detection should be disabled, and SID packets should only be transferred if they are required to keep the connection alive.

The decoding system of the network VHD 62 essentially performs the inverse operation of the encoding system. The decoding system of the network VHD 62 comprises a depacketizing engine 84, a voice queue 86, a DTMF queue 88, a precision tone queue 87, a voice synchronizer 90, a DTMF synchronizer 102, a precision tone synchronizer 103, a voice decoder 96, a VAD 98, a comfort noise estimator 100, a comfort noise generator 92, a lost packet recovery engine 94, a tone generator 104, and a precision tone generator 105.

The depacketizing engine 84 identifies the type of packets received from the host (i.e., voice packet, DTMF packet, call progress tone packet, SID packet), transforms them into frames which are protocol independent. The depacketizing engine 84 then transfers the voice frames (or voice parameters in the case of SID packets) into the voice queue 86, transfers the DTMF frames into the DTMF queue 88 and transfers the call progress tones into the call progress tone queue 87. In this manner, the remaining tasks are, by and large, protocol independent.

A jitter buffer is utilized to compensate for network impairments such as delay jitter caused by packets not arriving at the same time or in the same order in which they were transmitted. In addition, the jitter buffer compensates for lost packets that occur on occasion when the network is heavily congested. In the described exemplary embodiment, the jitter buffer for voice includes a voice synchronizer 90 that operates in conjunction with a voice queue 86 to provide an isochronous stream of voice frames to the voice decoder 96.

Sequence numbers embedded into the voice packets at the far end can be used to detect lost packets, packets arriving out of order, and short silence periods. The voice synchronizer 90 can analyze the sequence numbers, enabling the comfort noise generator 92 during short silence periods and performing voice frame repeats via the lost packet recovery engine 94 when voice packets are lost. SID packets can also be used as an indicator of silent periods causing the voice synchronizer 90 to enable the comfort noise generator 92. Otherwise, during far end active speech, the voice synchronizer 90 couples voice frames from the voice queue 86 in an isochronous stream to the voice decoder 96. The voice decoder 96 decodes the voice frames into digital voice samples suitable for transmission on a circuit switched network, such as a 64kb/s PCM signal for a PSTN line. The output of the voice decoder 96 (or the comfort noise generator 92 or lost packet recovery engine 94 if enabled) is written into a media queue 106 for transmission to the PXD 60.

The comfort noise generator 92 provides background noise to the near end user during silent periods. If the protocol supports SID packets, (and these are supported for VTOA, FRF-11, and VoIP), the comfort noise estimator at the far end encoding system should transmit SID packets. Then, the background noise can be reconstructed by the near end comfort noise generator 92 from the voice parameters in the SID packets buffered in the voice queue 86. However, for some protocols, namely, FRF-11, the SID packets are optional, and other far end users may not support SID packets at all. In these systems, the voice synchronizer 90 must continue to operate properly. In the absence of SID packets, the voice parameters of the

background noise at the far end can be determined by running the VAD 98 at the voice decoder 96 in series with a comfort noise estimator 100.

Preferably, the voice synchronizer 90 is not dependent upon sequence numbers embedded in the voice packet. The voice synchronizer 90 can invoke a number of mechanisms to compensate for delay jitter in these systems. For example, the voice synchronizer 90 can assume that the voice queue 86 is in an underflow condition due to excess jitter and perform packet repeats by enabling the lost frame recovery engine 94. Alternatively, the VAD 98 at the voice decoder 96 can be used to estimate whether or not the underflow of the voice queue 86 was due to the onset of a silence period or due to packet loss. In this instance, the spectrum and/or the energy of the digital voice samples can be estimated and the result 98a fed back to the voice synchronizer 90. The voice synchronizer 90 can then invoke the lost packet recovery engine 94 during voice packet losses and the comfort noise generator 92 during silent periods.

When DTMF packets arrive, they are depacketized by the depacketizing engine 84. DTMF frames at the output of the depacketizing engine 84 are written into the DTMF queue 88. The DTMF synchronizer 102 couples the DTMF frames from the DTMF queue 88 to the tone generator 104. Much like the voice synchronizer, the DTMF synchronizer 102 is employed to provide an isochronous stream of DTMF frames to the tone generator 104. Generally speaking, when DTMF packets are being transferred, voice frames should be suppressed. To some extent, this is protocol dependent. However, the capability to flush the voice queue 86 to ensure that the voice frames do not interfere with DTMF generation is desirable. Essentially, old voice frames which may be queued are discarded when DTMF packets arrive. This will ensure that there is a significant inter-digit gap before DTMF tones are generated. This is achieved by a "tone present" message 88a passed between the DTMF queue and the voice synchronizer 90.

The tone generator 104 converts the DTMF signals into a DTMF tone suitable for a standard digital or analog telephone. The tone generator 104 overwrites the media queue 106 to prevent leakage through the voice path and to ensure that the DTMF tones are not too noisy.

There is also a possibility that DTMF tone may be fed back as an echo into the DTMF detector 76. To prevent false detection, the DTMF detector 76 can be disabled entirely (or disabled only for the digit being generated) during DTMF tone generation. This is achieved by a “tone on” message 104a passed between the tone generator 104 and the DTMF detector 76. Alternatively, the NLP 72 can be activated while generating DTMF tones.

37367/CAG/B600

When call progress tone packets arrive, they are depacketized by the depacketizing engine 84. Call progress tone frames at the output of the depacketizing engine 84 are written into the call progress tone queue 87. The call progress tone synchronizer 103 couples the call progress tone frames from the call progress tone queue 87 to a call progress tone generator 105. Much like the DTMF synchronizer, the call progress tone synchronizer 103 is employed to provide an isochronous stream of call progress tone frames to the call progress tone generator 105. And much like the DTMF tone generator, when call progress tone packets are being transferred, voice frames should be suppressed. To some extent, this is protocol dependent. However, the capability to flush the voice queue 86 to ensure that the voice frames do not interfere with call progress tone generation is desirable. Essentially, old voice frames which may be queued are discarded when call progress tone packets arrive to ensure that there is a significant inter-digit gap before call progress tones are generated. This is achieved by a "tone present" message 87a passed between the call progress tone queue 87 and the voice synchronizer 90.

The call progress tone generator 105 converts the call progress tone signals into a call progress tone suitable for a standard digital or analog telephone. The call progress tone generator 105 overwrites the media queue 106 to prevent leakage through the voice path and to ensure that the call progress tones are not too noisy.

The outgoing PCM signal in the media queue 106 is coupled to the PXD 60 via the switchboard 32'. The outgoing PCM signal is coupled to an amplifier 108 before being outputted on the PCM output line 60b.

1. Echo Canceller with NLP

The problem of line echos such as the reflection of the talker's voice resulting from the operation of the hybrid with its two-four wire conversion is a common telephony problem. To eliminate or minimize the effect of line echos in the described exemplary embodiment of the present invention, an echo canceller with non-linear processing is used. Although echo cancellation is described in the context of a signal processing system for packet voice exchange, those skilled in the art will appreciate that the techniques described for echo cancellation are likewise suitable for various applications requiring the cancellation of reflections, or other undesirable signals, from a transmission line. Accordingly, the described exemplary embodiment for echo cancellation in a signal processing system is by way of example only and not by way of limitation.

37367/CAG/B600

In the described exemplary embodiment the echo canceller preferably complies with one or more of the following ITU-T Recommendations G.164 (1988) - Echo Suppressors, G.165 (March 1993) - Echo Cancellers, and G.168 (April 1997)- Digital Network Echo Cancellers, the contents of which are incorporated herein by reference as though set forth in full. The described embodiment merges echo cancellation and echo suppression methodologies to remove the line echos that are prevalent in telecommunication systems. Typically, echo cancellers are favored over echo suppressors for superior overall performance in the presence of system noise such as, for example, background music, double talk etc., while echo suppressors tend to perform well over a wide range of operating conditions where clutter such as system noise is not present. The described exemplary embodiment utilizes an echo suppressor when the energy level of the line echo is below the audible threshold, otherwise an echo canceller is preferably used. The use of an echo suppressor reduces system complexity, leading to lower overall power consumption or higher densities (more VHDs per part or network gateway). Those skilled in the art will appreciate that various signal characteristics such as energy, average magnitude, echo characteristics, as well as information explicitly received in voice or SID packets may be used to determine when to bypass echo cancellation. Accordingly, the described exemplary embodiment for bypassing echo cancellation in a signal processing system as a function of estimated echo power is by way of example only and not by way of limitation.

FIG. 11 shows the block diagram of an echo canceller in accordance with a preferred embodiment of the present invention. If required to support voice transmission via a T1 or other similar transmission media, a compressor 120 may compress the output 120(a) of the voice decoder system into a format suitable for the channel at R_{out} 120(b). Typically the compressor 120 provides μ-law or A-law compression in accordance with ITU-T standard G.711, although linear compression or compression in accordance with alternate companding laws may also be supported. The compressed signal at R_{out} (signal that eventually makes it way to a near end ear piece/telephone receiver), may be reflected back as an input signal to the voice encoder system. An input signal 122(a) may also be in the compressed domain (if compressed by compressor 120) and, if so, an expander 122 may be required to invert the companding law to obtain a near end signal 122(b). A power estimator 124 estimates a short term average power 124(a), a long term average power 124(b), and a maximum power level 124(c) for the near end signal 122(b).

An expander 126 inverts the companding law used to compress the voice decoder output signal 120(b) to obtain a reference signal 126(a). One of skill in the art will appreciated that the voice decoder output signal could alternatively be compressed downstream of the echo canceller

37367/CAG/B600

so that the expander 126 would not be required. However, to ensure that all non-linearities in the echo path are accounted for in the reference signal 126(a) it is preferable to compress / expand the voice decoder output signal 120(b). A power estimator 128 estimates a short term average power 128(a), a long term average power 128(b), a maximum power level 128(c) and a background power level 128(d) for the reference signal 126(a). The reference signal 126(a) is input into a finite impulse response (FIR) filter 130. The FIR filter 130 models the transfer characteristics of a dialed telephone line circuit so that the unwanted echo may preferably be canceled by subtracting filtered reference signal 130(a) from the near end signal 122(b) in a difference operator 132.

However, for a variety of reasons, such as for example, non-linearities in the hybrid and tail circuit, estimation errors, noise in the system, etc., the adaptive FIR filter 130 may not identically model the transfer characteristics of the telephone line circuit so that the echo canceller may be unable to cancel all of the resulting echo. Therefore, a non linear processor (NLP) 140 is used to suppress the residual echo during periods of far end active speech with no near end speech. During periods of inactive speech, a power estimator 138 estimates the performance of the echo canceller by estimating a short term average power 138(a), a long term average power 138(b) and background power level 138(c) for an error signal 132(b) which is an output of the difference operator 132. The estimated performance of the echo canceller is one measure utilized by adaptation logic 136 to selectively enable a filter adapter 134 which controls the convergence of the adaptive FIR filter 130. The adaptation logic 136 processes the estimated power levels of the reference signal (128a,128b,128c and 128d) the near end signal (124a,124b and 124c) and the error signal (138a, 138b and 138c) to control the invocation of the filter adapter 134 as well as the step size to be used during adaptation.

In the described preferred embodiment, the echo suppressor is a simple bypass 144(a) that is selectively enabled by toggling the bypass cancellation switch 144. A bypass estimator 142 toggles the bypass cancellation switch 144 based upon the maximum power level 128(c) of the reference signal 126(a), the long term average power 138(b) of the error signal 132(b) and the long term average power 124(b) of the near end signal 122(b). One skilled in the art will appreciate that a NLP or other suppressor could be included in the bypass path 144(a), so that the described echo suppressor is by way of example only and not by way of limitation.

In an exemplary embodiment, the adaptive filter 130 models the transfer characteristics of the hybrid and the tail circuit of the telephone circuit. The tail length supported should

37367/CAG/B600

preferably be at least 16 msec. The adaptive filter 130 may be a linear transversal filter or other suitable finite impulse response filter. In the described exemplary embodiment, the echo canceller preferably converges or adapts only in the absence of near end speech. Therefore, near end speech and/or noise present on the input signal 122(a) may cause the filter adapter 134 to diverge. To avoid divergence the filter adapter 134 is preferably selectively enabled by the adaptation logic 136. In addition, the time required for an adaptive filter to converge increases significantly with the number of coefficients to be determined. Reasonable modeling of the hybrid and tail circuits with a finite impulse response filter requires a large number of coefficients so that filter adaptation is typically computationally intense. In the described exemplary embodiment the DSP resources required for filter adaptation are minimized by adjusting the adaptation speed of the FIR filter 130.

The filter adapter 134 is preferably based upon a normalized least mean square algorithm (NLMS) as described in S. Haykin, Adaptive Filter Theory, and T. Parsons, Voice and Speech Processing, the contents of which are incorporated herein by reference as if set forth in full. The error signal 132(b) at the output of the difference operator 132 for the adaptation logic may preferably be characterized as follows:

$$e(n) = s(n) - \sum_{j=0}^{L-1} c(j)r(n-j)$$

where e(n) is the error signal at time n, r(n) is the reference signal 126(a) at time n and s(n) is the near end signal 122(b) at time n, and c(j) are the coefficients of the transversal filter where the dimension of the transversal filter is preferably the worst case echo path length (i.e. the length of the tail circuit L) and c(j), for j=0 to L-1, is given by:

$$c(j) = c(j) + \mu * e(n) * r(n-j)$$

wherein c(j) is preferably initialized to a reasonable value such as for example zero.

Assuming a block size of one msec (or 8 samples at a sampling rate of 8 kHz), the short term average power of the reference signal P_{ref} is the sum of the last L reference samples and the energy for the current eight samples so that

$$\mu = \frac{\alpha}{P_{ref(n)}}$$

37367/CAG/B600

where α is the adaptation step size. One of skill in the art will appreciate that the filter adaptation logic may be implemented in a variety of ways, including fixed point rather than the described floating point realization. Accordingly, the described exemplary adaptation logic is by way of example only and not by way of limitation.

To support filter adaptation the described exemplary embodiment includes the power estimator 128 that estimates the short term average power 128(a) of the reference signal 126(a) (P_{ref}). In the described exemplary embodiment the short term average power is preferably estimated over the worst case length of the echo path plus eight samples, (i.e. the length of the FIR filter L + 8 samples). In addition, the power estimator 128 computes the maximum power level 128(c) of the reference signal 126(a) (P_{refmax}) over a period of time that is preferably equal to the tail length L of the echo path. For example, putting a time index on the short term average power, so that $P_{ref}(n)$ is the power of the reference signal at time n. P_{refmax} is then characterized as:

$$P_{refmax}(n) = \max P_{ref}(j) \text{ for } j = n\text{-Lmsec to } j = n$$

where Lmsec is the length of the tail in msec so that P_{refmax} is the maximum power in the reference signal P_{ref} over a length of time equal to the tail length.

The second power estimator 124 estimates the short term average power of the near end signal 122(b) (P_{near}) in a similar manner. The short term average power 138(a) of the error signal 132(b) (the output of difference operator 132), P_{err} is also estimated in a similar manner by the third power estimator 138.

In addition, the echo return loss (ERL), defined as the loss from R_{out} 120(b) to S_{in} 122(a) in the absence of near end speech, is periodically estimated and updated. In the described exemplary embodiment the ERL is estimated and updated about every 5-20 msec. The power estimator 128 estimates the long term average power 128(b) (P_{refERL}) of the reference signal 126(a) in the absence of near end speech. The second power estimator 124 estimates the long term average power 124(b) ($P_{nearERL}$) of the near end signal 122(b) in the absence of near end speech. The adaptation logic 136 computes the ERL by dividing the long term average power of the reference signal (P_{refERL}) by the long term average power of the near end signal ($P_{nearERL}$). The adaptation logic 136 preferably only updates the long term averages used to compute the estimated ERL if the estimated short term power level 128(a) (P_{ref}) of the reference signal 126(a)

37367/CAG/B600

is greater than a predetermined threshold, preferably in the range of about -30 to -35 dBm0; and the estimated short term power level 128(a) (P_{ref}) of the reference signal 126(a) is preferably larger than about at least the short term average power 124(a) (P_{near}) of the near end signal 122(b) ($P_{ref} > P_{near}$ in the preferred embodiment).

In the preferred embodiment, the long term averages (P_{refERL} and $P_{nearERL}$)are based on a first order infinite impulse response (IIR) recursive filter, wherein the inputs to the two first order filters are P_{ref} and P_{near}.

$$P_{nearERL} = (1\text{-beta}) * P_{nearERL} + P_{near} * \text{beta; and}$$

$$P_{refERL} = (1\text{-beta}) * P_{refERL} + P_{ref} * \text{beta}$$

where filter coefficient beta = 1/64

Similarly, the adaptation logic 136 of the described exemplary embodiment characterizes the effectiveness of the echo canceller by estimating the echo return loss enhancement (ERLE). The ERLE is an estimation of the reduction in power of the near end signal 122(b) due to echo cancellation when there is no near end speech present. The ERLE is the average loss from the input 132(a) of the difference operator 132 to the output 132(b) of the difference operator 132. The adaptation logic 136 in the described exemplary embodiment periodically estimates and updates the ERLE, preferably in the range of about 5 to 20 msec. In operation, the power estimator 124 estimates the long term average power 124(b) $P_{nearERLE}$ of the near end signal 122(b) in the absence of near end speech. The power estimator 138 estimates the long term average power 138(b) $P_{errERLE}$ of the error signal 132(b) in the absence of near end speech. The adaptation logic 136 computes the ERLE by dividing the long term average power124(a) $P_{nearERLE}$ of the near end signal 122(b) by the long term average power 138(b) $P_{errERLE}$ of the error signal 132(b). The adaptation logic 136 preferably updates the long term averages used to compute the estimated ERLE only when the estimated short term average power 128(a) (P_{ref}) of the reference signal 126(a) is greater than a predetermined threshold preferably in the range of about -30 to -35 dBm0; and the estimated short term average power124(a) (P_{near}) of the near end signal 122(b) is large as compared to the estimated short term average power 138(a) (P_{err}) of the error signal (preferably when P_{near} is approximately greater than or equal to four times the short term average power of the error signal ($4P_{err}$)). Therefore, an ERLE of approximately 6 dB is preferably required before the ERLE tracker will begin to function.

37367/CAG/B600

In the preferred embodiment, the long term averages ($P_{nearERLE}$ and $P_{errERLE}$) may be based on a first order IIR (infinite impulse response) recursive filter, wherein the inputs to the two first order filters are P_{near} and P_{err}.

$$P_{nearERLE} = (1\text{-beta}) * P_{nearERL} + P_{near} * \text{beta; and}$$

$$P_{errERLE} = (1\text{-beta}) * P_{errERL} + P_{err} * \text{beta}$$

where filter coefficient beta = 1/64

It should be noted that PnearERL ≠ PnearERLE because the conditions under which each is updated are different.

To assist in the determination of whether to invoke the echo canceller and if so with what step size, the described exemplary embodiment estimates the power level of the background noise. The power estimator 128 tracks the long term energy level of the background noise 128(d) (B_{ref}) of the reference signal 126(a). The power estimator 128 utilizes a much faster time constant when the input energy is lower than the background noise estimate (current output). With a fast time constant the power estimator 128 tends to track the minimum energy level of the reference signal 126(a). By definition, this minimum energy level is the energy level of the background noise of the reference signal B_{ref}. The energy level of the background noise of the error signal B_{err} is calculated in a similar manner. The estimated energy level of the background noise of the error signal (B_{err}) is not updated when the energy level of the reference signal is larger than a predetermined threshold (preferably in the range of about 30-35 dBm0).

In addition, the invocation of the echo canceller depends on whether near end speech is active. Preferably, the adaptation logic 136 declares near end speech active when three conditions are met. First, the short term average power of the error signal should preferably exceed a minimum threshold, preferably on the order of about -36 dBm0 ($P_{err} \geq -36$ dBm0). Second, the short term average power of the error signal should preferably exceed the estimated power level of the background noise for the error signal by preferably at least about 6 dB ($P_{err} \geq B_{err} + 6$ dB). Third, the short term average power 124(a) of the near end signal 122(b) is preferably approximately 3 dB greater than the maximum power level 128(c) of the reference signal 126(a) less the estimated ERL ($P_{near} \geq P_{refmax}$ - ERL + 3dB). The adaptation logic 136 preferably sets a near end speech hangover counter (not shown) when near end speech is

37367/CAG/B600

detected. The hangover counter is used to prevent clipping of near end speech by delaying the invocation of the NLP 140 when near end speech is detected. Preferably the hangover counter is on the order of about 150 msec.

In the described exemplary embodiment, if the maximum power level (P_{refmax}) of the reference signal minus the estimated ERL is less than the threshold of hearing (all in dB) neither echo cancellation or non-linear processing are invoked. In this instance, the energy level of the echo is below the threshold of hearing, typically about -65 to -69 dBm0, so that echo cancellation and non-linear processing are not required for the current time period. Therefore, the bypass estimator 142 sets the bypass cancellation switch 144 in the down position, so as to bypass the echo canceller and the NLP and no processing (other than updating the power estimates) is performed. Also, if the maximum power level (P_{refmax}) of the reference signal minus the estimated ERL is less than the maximum of either the threshold of hearing, or background power level B_{err} of the error signal minus a predetermined threshold (P_{refmax} - ERL < threshold of hearing or (B_{err} - threshold)) neither echo cancellation or non-linear processing are invoked. In this instance, the echo is buried in the background noise or below the threshold of hearing, so that echo cancellation and non-linear processing are not required for the current time period. In the described preferred embodiment the background noise estimate is preferably greater than the threshold of hearing, such that this is a broader method for setting the bypass cancellation switch. The threshold is preferably in the range of about 8-12 dB.

Similarly, if the maximum power level (P_{refmax}) of the reference signal minus the estimated ERL is less than the short term average power P_{near} minus a predetermined threshold (P_{refmax} - ERL < P_{near} - threshold) neither echo cancellation or non-linear processing are invoked. In this instance, it is highly probable that near end speech is present, and that such speech will likely mask the echo. This method operates in conjunction with the above described techniques for bypassing the echo canceller and NLP. The threshold is preferably in the range of about 8-12 dB. If the NLP contains a real comfort noise generator, i.e., a non-linearity which mutes the incoming signal and injects comfort noise of the appropriate character then a determination that the NLP will be invoked in the absence of filter adaptation allows the adaptive filter to be bypassed or not invoked. This method is, used in conjunction with the above methods. If the adaptive filter is not executed then adaptation does not take place, so this method is preferably used only when the echo canceller has converged.

37367/CAG/B600

If the bypass cancellation switch 144 is in the down position, the adaptation logic 136 disables the filter adapter 134. Otherwise, for those conditions where the bypass cancellation switch 144 is in the up position so that both adaptation and cancellation may take place, the operation of the preferred adaptation logic 136 proceeds as follows:

If the estimated echo return loss enhancement is low (preferably in the range of about 0-9dBm) the adaptation logic 136 enables rapid convergence with an adaptation step size $\alpha = 1/4$. In this instance , the echo canceller is not converged so that rapid adaptation is warranted. However, if near end speech is detected within the hangover period, the adaptation logic 136 either disables adaptation or uses very slow adaptation, preferably an adaptation speed on the order of about one-eighth that used for rapid convergence or an adaptation step size $\alpha = 1/32$. In this case the adaptation logic 136 disables adaptation when the echo canceller is converged. Convergence may be assumed if adaptation has been active for a total of one second after the off hook transition or subsequent to the invocation of the echo canceller. Otherwise if the combined loss (ERL+ERLE) is in the range of about 33-36 dB, the adaptation logic 136 enables slow adaptation (preferably one-eighth the adaptation speed of rapid convergence or an adaptation step size $\alpha=1/32$). If the combined loss (ERL+ERLE) is in the range of about 23-33 dB, the adaptation logic 136 enables a moderate convergence speed, preferably on the order of about one-fourth the adaptation speed used for rapid convergence or an adaptation step size $\alpha = 1/16$.

Otherwise, one of three preferred adaptation speeds is chosen based on the estimated echo power (P_{refmax} minus the ERL) in relation to the power level of the background noise of the error signal. If the estimated echo power (P_{refmax} - ERL) is large compared to the power level of the background noise of the error signal (P_{refmax} - ERL $\geq B_{err}$+24 dB), rapid adaptation / convergence is enabled with an adaptation step size on the order of about $\alpha = 1/4$. Otherwise, if (P_{refmax} - ERL $\geq B_{err}$ + 18 dB) the adaptation speed is reduced to approximately one-half the adaptation speed used for rapid convergence or an adaptation step size on the order of about $\alpha = 1/8$. Otherwise, if (P_{refmax} - ERL $\geq B_{err}$ + 9 dB) the adaptation speed is further reduced to approximately one-quarter the adaptation speed used for rapid convergence or an adaptation step size $\alpha = 1/16$.

As a further limit on adaptation speed, if echo canceller adaptation has been active for a sum total of one second since initialization or an off-hook condition then the maximum adaptation speed is limited to one-fourth the adaptation speed used for rapid convergence ($\alpha=1/16$). Also, if the echo path changes appreciably or if for any reason the estimated ERLE is negative, (which typically occurs when the echo path changes) then the coefficients are cleared

37367/CAG/B600

and an adaptation counter is set to zero (the adaptation counter measures the sum total of adaptation cycles in samples).

The NLP 140 is a two state device. The NLP 140 is either on (applying non-linear processing) or it is off (applying unity gain). When the NLP 140 is on it tends to stay on, and when the NLP 140 is off it tends to stay off. The NLP 140 is preferably invoked when the bypass cancellation switch 144 is in the upper position so that adaptation and cancellation are active. Otherwise, the NLP 140 is not invoked and the NLP 140 is forced into the off state.

Initially, a stateless first NLP decision is created. The decision logic is based on three decision variables (D1- D3). The decision variable D1 is set if it is likely that the far end is active (i.e. the short term average power 128(a) of the reference signal 126(a) is preferably about 6 dB greater than the power level of the background noise 128(d) of the reference signal), and the short term average power 128(a) of the reference signal 126(a) minus the estimated ERL is greater than the estimated short term average power 124(a) of the near end signal 122(b) minus a small threshold, preferably in the range of about 6 dB. In the preferred embodiment, this is represented by: ($P_{ref} \geq B_{ref} + 6$ dB) and ((P_{ref} - ERL) $\geq$ (P_{near} - 6 dB)). Thus, decision variable D1 attempts to detect far end active speech and high ERL (implying no near end). Preferably, decision variable D2 is set if the power level of the error signal is on the order of about 9 dB below the power level of the estimated short term average power 124(a) of the near end signal 122(b) (a condition that is indicative of good short term ERLE). In the preferred embodiment, $P_{err} \leq P_{near}$ - 9 dB is used (a short term ERLE of 9 dB). The third decision variable D3 is preferably set if the combined loss (reference power to error power) is greater than a threshold. In the preferred embodiment, this is: $P_{err} \leq P_{ref} - t$, where t is preferably initialized to about 6 dB and preferably increases to about 12 dB after about one second of adaptation. (In other words, it is only adapted while convergence is enabled).

The third decision variable D3 results in more aggressive non linear processing while the echo canceller is uncovered. Once the echo canceller converges, the NLP 140 can be slightly less aggressive. The initial stateless decision is set if two of the sub-decisions or control variables are initially set. The initial decision set implies that the NLP 140 is in a transition state or remaining on.

A NLP state machine (not shown) controls the invocation and termination of NLP 140 in accordance with the detection of near end speech as previously described. The NLP state

37367/CAG/B600

machine delays activation of the NLP 140 when near end speech is detected to prevent clipping the near end speech. In addition, the NLP state machine is sensitive to the near end speech hangover counter (set by the adaptation logic when near end speech is detected) so that activation of the NLP 140 is further delayed until the near end speech hangover counter is cleared. The NLP state machine also deactivates the NLP 140. The NLP state machine preferably sets an off counter when the NLP 140 has been active for a predetermined period of time, preferably about the tail length in msec. The "off" counter is cleared when near end speech is detected and decremented while non-zero when the NLP is on. The off counter delays termination of NLP processing when the far end power decreases so as to prevent the reflection of echo stored in the tail circuit. If the near end speech detector hangover counter is on, the above NLP decision is overridden and the NLP is forced into the off state.

In the preferred embodiment, the NLP 140 may be implemented with a suppressor that adaptively suppresses down to the background noise level (B_{err}), or a suppressor that suppresses completely and inserts comfort noise with a spectrum that models the true background noise.

2. Automatic Gain Control

In an exemplary embodiment of the present invention, AGC is used to normalize digital voice samples to ensure that the conversation between the near and far end users is maintained at an acceptable volume. The described exemplary embodiment of the AGC includes a signal bypass for the digital voice samples when the gain adjusted digital samples exceeds a predetermined power level. This approach provides rapid response time to increased power levels by coupling the digital voice samples directly to the output of the AGC until the gain falls off due to AGC adaptation. Although AGC is described in the context of a signal processing system for packet voice exchange, those skilled in the art will appreciate that the techniques described for AGC are likewise suitable for various applications requiring a signal bypass when the processing of the signal produces undesirable results. Accordingly, the described exemplary embodiment for AGC in a signal processing system is by way of example only and not by way of limitation.

In an exemplary embodiment, the AGC can be either fully adaptive or have a fixed gain. Preferably, the AGC supports a fully adaptive operating mode with a range of about -30 dB to 30 dB. A default gain value may be independently established, and is typically 0 dB. If adaptive gain control is used, the initial gain value is specified by this default gain. The AGC adjusts the

37367/CAG/B600

gain factor in accordance with the power level of an input signal. Input signals with a low energy level are amplified to a comfortable sound level, while high energy signals are attenuated.

A block diagram of a preferred embodiment of the AGC is shown in FIG. 12A. A multiplier 150 applies a gain factor 152 to an input signal 150(a) which is then output to the media queue 66 of the network VHD (see FIG. 10). The default gain, typically 0 dB is initially applied to the input signal 150(a). A power estimator 154 estimates the short term average power 154(a) of the gain adjusted signal 150(b). The short term average power of the input signal 150(a) is preferably calculated every eight samples, typically every one ms for a 8 kHz signal. Clipping logic 156 analyzes the short term average power 154(a) to identify gain adjusted signals 150(b) whose amplitudes are greater than a predetermined clipping threshold. The clipping logic 156 controls an AGC bypass switch 157, which directly connects the input signal 150(a) to the media queue 66 when the amplitude of the gain adjusted signal 150(b) exceeds the predetermined clipping threshold. The AGC bypass switch 157 remains in the up or bypass position until the AGC adapts so that the amplitude of the gain adjusted signal 150(b) falls below the clipping threshold.

The power estimator 154 also calculates a long term average power 154(b) for the input signal 150(a), by averaging thirty two short term average power estimates, (i.e. averages thirty two blocks of eight samples). The long term average power is a moving average which provides significant hangover. A peak tracker 158 utilizes the long term average power 154(b) to calculate a reference value which gain calculator 160 utilizes to estimate the required adjustment to a gain factor 152. The gain factor 152 is applied to the input signal 150(a) by the multiplier 150. In the described exemplary embodiment the peak tracker 158 may preferably be a non-linear filter. The peak tracker 158 preferably stores a reference value which is dependent upon the last maximum peak. The peak tracker 158 compares the long term average power estimate to the reference value. FIG. 12B shows the peak tracker output as a function of an input signal, demonstrating that the reference value that the peak tracker 158 forwards to the gain calculator 160 should preferably rise quickly if the signal amplitude increases, but decrement slowly if the signal amplitude decreases. Thus for active voice segments followed by silence, the peak tracker output slowly decreases, so that the gain factor applied to the input signal 150(a) may be slowly increased. However, for long inactive or silent segments followed by loud or high amplitude voice segments, the peak tracker output increases rapidly, so that the gain factor applied to the input signal 150(a) may be quickly decreased.

37367/CAG/B600

In the described exemplary embodiment, the peak tracker should be updated when the estimated long term power exceeds the threshold of hearing. Peak tracker inputs include the current estimated long term power level a(i), the previous long term power estimate, a(i-1), and the previous peak tracker output x(i-1). In operation, when the long term energy is varying rapidly, preferably when the previous long term power estimate is on the order of four times greater than the current long term estimate or vice versa, the peak tracker should go into hangover mode. In hangover mode, the peak tracker should not be updated. The hangover mode prevents adaptation on impulse noise.

If the long term energy estimate is large compared to the previous peak tracker estimate, then the peak tracker should adapt rapidly. In this case the current peak tracker output x(i) is given by:

$$x(i) = (7x(i-1) + a(i))/8.$$

where x(i-1) is the previous peak tracker output and a(i) is the current long term power estimate.

If the long term energy is less than the previous peak tracker output, then the peak tracker will adapt slowly. In this case the current peak tracker output x(i) is given by:

$$x(i) = x(i-1) * 255/256.$$

Referring to FIG. 13, a preferred embodiment of the gain calculator 160 slowly increments the gain factor 152 for signals below the comfort level of hearing 166 (below minVoice) and decrements the gain for signals above the comfort level of hearing 164 (above MaxVoice). The described exemplary embodiment of the gain calculator 160 decrements the gain factor 152 for signals above the clipping threshold relatively fast, preferably on the order of about 2-4 dB/sec, until the signal has been attenuated approximately 10 dB or the power level of the signal drops to the comfort zone. The gain calculator 160 preferably decrements the gain factor 152 for signals with power levels that are above the comfort level of hearing 164 (MaxVoice) but below the clipping threshold 166 (Clip) relatively slowly, preferably on the order of about 0.1-0.3 dB/sec until the signal has been attenuated approximately 4 dB or the power level of the signal drops to the comfort zone.

37367/CAG/B600

The gain calculator 160 preferably does not adjust the gain factor 152 for signals with power levels within the comfort zone (between minVoice and MaxVoice), or below the maximum noise power threshold 168 (MaxNoise). The preferred values of MaxNoise, min Voice, MaxVoice, Clip are related to a noise floor 170 and are preferably in 3dB increments. The noise floor is preferably empirically derived by calibrating the host DSP platform with a known load. The noise floor preferably adjustable and is typically within the range of about, -45 to -52 dBm. A MaxNoise value of two corresponds to a power level 6 dB above the noise floor 170, whereas a clip level of nine corresponds to 27 dB above noise floor 170. For signals with power levels below the comfort zone (less than minVoice) but above the maximum noise threshold, the gain calculator 160 preferably increments the gain factor 152 logarithmically at a rate of about 0.1-0.3 dB/sec, until the power level of the signal is within the comfort zone or a gain of approximately 10 dB is reached.

In the described exemplary embodiment, the AGC is designed to adapt slowly, although it should adapt fairly quickly if overflow or clipping is detected. From a system point of view, AGC adaptation should be held fixed if the NLP 72 (see FIG. 10) is activated or the VAD 80 (see FIG. 10) determines that voice is inactive. In addition, the AGC is preferably sensitive to the amplitude of received call progress tones. In the described exemplary embodiment, rapid adaptation may be enabled as a function of the actual power level of a received call progress tone such as for example a ring back tone, compared to the power levels set forth in the applicable standards.

3. Voice Activity Detector

In an exemplary embodiment, the VAD, in either the encoder system or the decoder system, can be configured to operate in multiple modes so as to provide system tradeoffs between voice quality and bandwidth requirements. In a first mode, the VAD is always disabled and declares all digital voice samples as active speech. This mode is applicable if the signal processing system is used over a TDM network, a network which is not congested with traffic, or when used with PCM (ITU Recommendation G.711 (1988) - Pulse Code Modulation (PCM) of Voice Frequencies, the contents of which is incorporated herein by reference as if set forth in full) in a PCM bypass mode for supporting data or fax modems.

In a second "transparent" mode, the voice quality is indistinguishable from the first mode. In transparent mode, the VAD identifies digital voice samples with an energy below the

threshold of hearing as inactive speech. The threshold may be adjustable between -90 and - 40 dBm with a default value of - 60 dBm. The transparent mode may be used if voice quality is much more important than bandwidth. This may be the case, for example, if a G.711 voice encoder (or decoder) is used.

In a third "conservative" mode, the VAD identifies low level (but audible) digital voice samples as inactive, but will be fairly conservative about discarding the digital voice samples. A low percentage of active speech will be clipped at the expense of slightly higher transmit bandwidth. In the conservative mode, a skilled listener may be able to determine that voice activity detection and comfort noise generation is being employed. The threshold for the conservative mode may preferably be adjustable between -65 and - 35 dBm with a default value of - 60 dBm.

In a fourth "aggressive" mode, bandwidth is at a premium. The VAD is aggressive about discarding digital voice samples which are declared inactive. This approach will result in speech being occasionally clipped, but system bandwidth will be vastly improved. The threshold for the aggressive mode may preferably be adjustable between -60 and - 30 dBm with a default value of - 55 dBm.

The transparent mode is typically the default mode when the system is operating with 16 bit PCM, companded PCM (G.711) or adaptive differential PCM (ITU Recommendations G.726 (Dec. 1990) - 40, 32, 24, 16 kbit/s Using Low-Delay Code Exited Linear Prediction, and G.727 (Dec. 1990) - 5 -, 4 -, 3 -, and 2 - Sample Embedded Adaptive Differential Pulse Code Modulation). In these instances, the user is most likely concerned with high quality voice since a high bit-rate voice encoder (or decoder) has been selected. As such, a high quality VAD should be employed. The transparent mode should also be used for the VAD operating in the decoder system since bandwidth is not a concern (the VAD in the decoder system is used only to update the comfort noise parameters). The conservative mode could be used with ITU Recommendation G.728 (Sept. 1992) - Coding of Speech at 16 kbit/s Using Low-Delay Code Excited Linear Prediction, G.729, and G.723.1. For systems demanding high bandwidth efficiency, the aggressive mode can be employed as the default mode.

The mechanism in which the VAD detects digital voice samples that do not contain active speech can be implemented in a variety of ways. One such mechanism entails monitoring the energy level of the digital voice samples over short periods (where a period length is typically

37367/CAG/B600

in the range of about 10 to 30 msec). If the energy level exceeds a fixed threshold, the digital voice samples are declared active, otherwise they are declared inactive. The transparent mode can be obtained when the threshold is set to the threshold level of hearing.

Alternatively, the threshold level of the VAD can be adaptive and the background noise energy can be tracked. If the energy in the current period is sufficiently larger than the background noise estimate by the comfort noise estimator, the digital voice samples are declared active, otherwise they are declared inactive. The VAD may also freeze the comfort noise estimator or extend the range of active periods (hangover). This type of VAD is used in GSM (European Digital Cellular Telecommunications System; Half rate Speech Part 6: Voice Activity Detector (VAD) for Half Rate Speech Traffic Channels (GSM 6.42), the contents of which is incorporated herein by reference as if set forth in full) and QCELP (W. Gardner, P. Jacobs, and C. Lee, "QCELP: A Variable Rate Speech Coder for CDMA Digital Cellular," in *Speech and Audio Coding for Wireless and Network Applications*, B.S. atal, V. Cuperman, and A. Gersho (eds)., the contents of which is incorporated herein by reference as if set forth in full).

In a VAD utilizing an adaptive threshold level, speech parameters such as the zero crossing rate, spectral tilt, energy and spectral dynamics are measured and compared to stored values for noise. If the parameters differ significantly from the stored values, it is an indication that active speech is present even if the energy level of the digital voice samples is low.

When the VAD operates in the conservative or transparent mode, measuring the energy of the digital voice samples can be sufficient for detecting inactive speech. However, the spectral dynamics of the digital voice samples against a fixed threshold may be useful in discriminating between long voice segments with audio spectra and long term background noise. In an exemplary embodiment of a VAD employing spectral analysis, the VAD performs auto-correlations using Itakura or Itakura-Saito distortion to compare long term estimates based on background noise to short term estimates based on a period of digital voice samples. In addition, if supported by the voice encoder, line spectrum pairs (LSPs) can be used to compare long term LSP estimates based on background noise to short terms estimates based on a period of digital voice samples. Alternatively, FFT methods can be are used when the spectrum is available from another software module.

Preferably, hangover should be applied to the end of active periods of the digital voice samples with active speech. Hangover bridges short inactive segments to ensure that quiet

trailing, unvoiced sounds (such as /s/), are classified as active. The amount of hangover can be adjusted according to the mode of operation of the VAD. If a period following a long active period is clearly inactive (i.e., very low energy with a spectrum similar to the measured background noise) the length of the hangover period can be reduced. Generally, a range of about 40 to 300 msec of inactive speech following an active speech burst will be declared active speech due to hangover.

4. Comfort Noise Generator

According to industry research the average voice conversation includes as much as sixty percent silence or inactive content so that transmission across the packet based network can be significantly reduced if non-active speech packets are not transmitted across the packet based network. In an exemplary embodiment of the present invention, a comfort noise generator is used to effectively reproduce background noise when non-active speech packets are not received. In the described preferred embodiment. comfort noise is generated as a function signal characteristics received from a remote source and estimated signal characteristics. In the described exemplary embodiment comfort noise parameters are preferably generated by a comfort noise estimator. The comfort noise parameters may be transmitted from the far end or can be generated by monitoring the energy level and spectral characteristics of the far end noise at the end of active speech (i.e., during the hangover period). Although comfort noise generation is described in the context of a signal processing system for packet voice exchange, those skilled in the art will appreciate that the techniques described for comfort noise generation are likewise suitable for various applications requiring reconstruction of a signal from signal parameters. Accordingly, the described exemplary embodiment for comfort noise generation in a signal processing system for voice applications is by way of example only and not by way of limitation.

A comfort noise generator plays noise. In an exemplary embodiment, a comfort noise generator in accordance with ITU standards G.729 Annex B or G.723.1 Annex A may be used. These standards specify background noise levels and spectral content. Referring to FIG. 10, the VAD 80 in the encoder system determines whether the digital voice samples in the media queue 66 contain active speech. If the VAD 80 determines that the digital voice samples do not contain active speech, then the comfort noise estimator 81 estimates the energy and spectrum of the background noise parameters at the near end to update a long running background noise energy and spectral estimates. These estimates are periodically quantized and transmitted in a SID packet by the comfort noise estimator (usually at the end of a talk spurt and periodically during

37367/CAG/B600

the ensuing silent segment, or when the background noise parameters change appreciably). The comfort noise estimator 81 should update the long running averages, when necessary, decide when to transmit a SID packet, and quantize and pass the quantized parameters to the packetization engine 78. SID packets should not be sent while the near end telephony device is on-hook, unless they are required to keep the connection between the telephony devices alive. There may be multiple quantization methods depending on the protocol chosen.

In many instances the characterization of spectral content or energy level of the background noise may not be available to the comfort noise generator in the decoder system. For example, SID packets may not be used or the contents of the SID packet may not be specified (see FRF-11). Similarly, the SID packets may only contain an energy estimate, so that estimating some or all of the parameters of the noise in the decoding system may be necessary. Therefore, the comfort noise generator 92 (see FIG.11) preferably should not be dependent upon SID packets from the far end encoder system for proper operation.

In the absence of SID packets, or SID packets containing energy only, the parameters of the background noise at the far end may be estimated by either of two alternative methods. First, the VAD 98 at the voice decoder 96 can be executed in series with the comfort noise estimator 100 to identify silence periods and to estimate the parameters of the background noise during those silence periods. During the identified inactive periods, the digital samples from the voice decoder 96 are used to update the comfort noise parameters of the comfort noise estimator. The far end voice encoder should preferably ensure that a relatively long hangover period is used in order to ensure that there are noise-only digital voice samples which the VAD 98 may identify as inactive speech.

Alternatively, in the case of SID packets containing energy levels only, the comfort noise estimate may be updated with the two or three digital voice frames which arrived immediately prior to the SID packet. The far end voice encoder should preferably ensure that at least two or three frames of inactive speech are transmitted before the SID packet is transmitted. This can be realized by extending the hangover period. The comfort noise estimator 100 may then estimate the parameters of the background noise based upon the spectrum and or energy level of these frames. In this alternate approach continuous VAD execution is not required to identify silence periods, so as to further reduce the average bandwidth required for a typical voice channel.

37367/CAG/B600

Alternatively, if it is unknown whether or not the far end voice encoder supports (sending) SID packets, the decoder system may start with the assumption that SID packets are not being sent, utilizing a VAD to identify silence periods, and then only use the comfort noise parameters contained in the SID packets if and when a SID packet arrives.

A preferred embodiment of the comfort noise generator generates comfort noise based upon the energy level of the background noise contained within the SID packets and spectral information derived from the previously decoded inactive speech frames. The described exemplary embodiment (in the decoding system) includes a comfort noise estimator for noise analysis and a comfort noise generator for noise synthesis. Preferably there is an extended hangover period during which the decoded voice samples is primarily inactive before the VAD identifies the signal as being inactive, (changing from speech to noise). Linear Prediction Coding (LPC) coefficients may be used to model the spectral shape of the noise during the hangover period just before the SID packet is received from the VAD. Linear prediction coding models each voice sample as a linear combination of previous samples, that is, as the output of an all-pole IIR filter. Referring to FIG. 14, a noise analyzer 174 determines the LPC coefficients.

In the described exemplary embodiment of the comfort noise estimator in the decoding system, a signal buffer 176 receives and buffers decoded voice samples. An energy estimator 177 analyzes the energy level of the samples buffered in the signal buffer 176. The energy estimator 177 compares the estimated energy level of the samples stored in the signal buffer with the energy level provided in the SID packet. Comfort noise estimating is terminated if the energy level estimated for the samples stored in the signal buffer and the energy level provided in the SID packet differ by more than a predetermined threshold, preferably on the order of about 6 dB. In addition, the energy estimator 177, analyzes the stability of the energy level of the samples buffered in the signal buffer. The energy estimator 177 preferably divides the samples stored in the signal buffer into two groups, (preferably approximately equal halves) and estimates the energy level for each group. Comfort noise estimation is preferably terminated if the estimated energy levels of the two groups differ by more than a predetermined threshold, preferably on the order of about 6 dB. A shaping filter 178 filters the incoming voice samples from the energy estimator 177 with a triangular windowing technique. Those of skill in the art will appreciate that alternative shaping filters such as, for example, a Hamming window, may be used to shape the incoming samples.

37367/CAG/B600

When a SID packet is received in the decoder system, auto correlation logic 179 calculates the auto-correlation coefficients of the windowed voice samples. The signal buffer 176 should preferably be sized to be smaller than the hangover period, to ensure that the auto correlation logic 179 computes auto correlation coefficients using only voice samples from the hangover period. In the described exemplary embodiment, the signal buffer is sized to store on the order of about two hundred voice samples (25 msec assuming a sample rate of 8000 Hz). Autocorrelation, as is known in the art, involves correlating a signal with itself. A correlation function shows how similar two signals are and how long the signals remain similar when one is shifted with respect to the other. Random noise is defined to be uncorrelated, that is random noise is only similar to itself with no shift at all. A shift of one sample results in zero correlation, so that the autocorrelation function of random noise is a single sharp spike at shift zero. The autocorrelation coefficients are calculated according to the following equation:

$$r(k) = \sum_{n=k}^{m} s(n)s(n-k)$$

where k=0...p and p is the order of the synthesis filter 188 (see FIG. 15) utilized to synthesize the spectral shape of the background noise from the LPC filter coefficients.

Filter logic 180 utilizes the auto correlation coefficients to calculate the LPC filter coefficients 180(a) and prediction gain 180(b) using the Levinson-Durbin Recursion method. Preferrably, the filter logic 180 first preferably applies a white noise correction factor to r(0) to increase the energy level of r(0) by a predetermined amount. The preferred white noise correction factor is on the order of about (257/256) which corresponds to a white noise level of approximately 24 dB below the average signal power. The white noise correction factor effectively raises the spectral minima so as to reduce the spectral dynamic range of the auto correlation coefficients to alleviate ill-conditioning of the Levinson-Durbin recursion. As is known in the art, the Levinson-Durbin recursion is an algorithm for finding an all-pole IIR filter with a prescribed deterministic autocorrelation sequence. The described exemplary embodiment preferably utilizes a tenth order (i.e. ten tap) synthesis filter 188. However, a lower order filter may be used to realize a reduced complexity comfort noise estimator.

The signal buffer 176 should preferably be updated each time the voice decoder is invoked during periods of active speech. Therefore, when there is a transition from speech to noise, the buffer 176 contains the voice samples from the most recent hangover period. The comfort noise estimator should preferably ensure that the LPC filter coefficients is determined

37367/CAG/B600

using only samples of background noise. If the LPC filter coefficients are determined based on the analysis of active speech samples, the estimated LPC filter coefficients will not give the correct spectrum of the background noise. In the described exemplary embodiment, a hangover period in the range of about 50-250 msec is assumed, and twelve active frames (assuming 5 msec frames) are accumulated before the filter logic 180 calculates new LPC coefficients.

In the described exemplary embodiment a comfort noise generator utilizes the power level of the background noise retrieved from processed SID packets and the predicted LPC filter coefficients 180(a) to generate comfort noise in accordance with the following formula:

$$s(n) = e(n) + \sum_{i=1}^{M} a(i)s(n-i)$$

Where M is the order (i.e. the number of taps) of the synthesis filter 188, s(n) is the predicted value of the synthesized noise, a(i) is the i^{th} LPC filter coefficient, s(n-i) are the previous output samples of the synthesis filter and e(n) is a Gaussian excitation signal.

A block diagram of the described exemplary embodiment of the comfort noise generator 182 is shown in FIG. 15. The comfort noise estimator processes SID packets to decode the power level of the current far end background noise. The power level of the background noise is forwarded to a power controller 184. In addition a white noise generator 186 forwards a gaussian signal to the power controller 184. The power controller 184 adjusts the power level of the gaussian signal in accordance with the power level of the background noise and the prediction gain 180(b). The prediction gain is the difference in power level of the input and output of synthesis filter 188. The synthesis filter 188 receives voice samples from the power controller 184 and the LPC filter coefficients calculated by the filter logic 180 (see FIG. 14). The synthesis filter 188 generates a power adjusted signal whose spectral characteristics approximate the spectral shape of the background noise in accordance with the above equation (i.e. sum of the product of the LPC filter coefficients and the previous output samples of the synthesis filter).

5. Voice Encoder/Voice Decoder

The purpose of voice compression algorithms is to represent voice with highest efficiency (i.e., highest quality of the reconstructed signal using the least number of bits). Efficient voice compression was made possible by research starting in the 1930's that demonstrated that voice could be characterized by a set of slowly varying parameters that could later be used to

reconstruct an approximately matching voice signal. Characteristics of voice perception allow for lossy compression without perceptible loss of quality.

Voice compression begins with an analog-to-digital converter that samples the analog voice at an appropriate rate (usually 8,000 samples per second for telephone bandwidth voice) and then represents the amplitude of each sample as a binary code that is transmitted in a serial fashion. In communications systems, this coding scheme is called pulse code modulation (PCM).

When using a uniform (linear) quantizer in which there is uniform separation between amplitude levels. This voice compression algorithm is referred to as "linear", or "linear PCM". Linear PCM is the simplest and most natural method of quantization. The drawback is that the signal-to-noise ratio (SNR) varies with the amplitude of the voice sample. This can be substantially avoided by using non-uniform quantization known as companded PCM..

In companded PCM, the voice sample is compressed to logarithmic scale before transmission, and expanded upon reception. This conversion to logarithmic scale ensures that low-amplitude voice signals are quantized with a minimum loss of fidelity, and the SNR is more uniform across all amplitudes of the voice sample. The process of compressing and expanding the signal is known as "companding" (COMpressing and exPANDing). There exists a worldwide standard for companded PCM defined by the CCITT (the International Telegraph and Telephone Consultative Committee).

The CCITT is a Geneva-based division of the International Telecommunications Union (ITU), a New York-based United Nations organization. The CCITT is now formally known as the ITU-T, the telecommunications sector of the ITU, but the term CCITT is still widely used. Among the tasks of the CCITT is the study of technical and operating issues and releasing recommendations on them with a view to standardizing telecommunications on a worldwide basis. A subset of these standards is the G-Series Recommendations, which deal with the subject of transmission systems and media, and digital systems and networks. Since 1972, there have been a number of G-Series Recommendations on speech coding, the earliest being Recommendation G.711. G.711 has the best voice quality of the compression algorithms but the highest bit rate requirement.

37367/CAG/B600

The ITU-T defined the "first" voice compression algorithm for digital telephony in 1972. It is companded PCM defined in Recommendation G.711. This Recommendation constitutes the principal reference as far as transmission systems are concerned. The basic principle of the G.711 companded PCM algorithm is to compress voice using 8 bits per sample, the voice being sampled at 8 kHz, keeping the telephony bandwidth of 300-3400 Hz. With this combination, each voice channel requires 64 kilobits per second.

Note that when the term PCM is used in digital telephony, it usually refers to the companded PCM specified in Recommendation G.711, and not linear PCM, since most transmission systems transfer data in the companded PCM format. Companded PCM is currently the most common digitization scheme used in telephone networks. Today, nearly every telephone call in North America is encoded at some point along the way using G.711 companded PCM.

ITU Recommendation G.726 specifies a multiple-rate ADPCM compression technique for converting 64 kilobit per second companded PCM channels (specified by Recommendation G.711) to and from a 40, 32, 24, or 16 kilobit per second channel. The bit rates of 40, 32, 24, and 16 kilobits per second correspond to 5, 4, 3, and 2 bits per voice sample.

ADPCM is a combination of two methods: Adaptive Pulse Code Modulation (APCM), and Differential Pulse Code Modulation (DPCM). Adaptive Pulse Code Modulation can be used in both uniform and non-uniform quantizer systems. It adjusts the step size of the quantizer as the voice samples change, so that variations in amplitude of the voice samples, as well as transitions between voiced and unvoiced segments, can be accommodated. In DPCM systems, the main idea is to quantize the difference between contiguous voice samples. The difference is calculated by subtracting the current voice sample from a signal estimate predicted from previous voice sample. This involves maintaining an adaptive predictor (which is linear, since it only uses first-order functions of past values). The variance of the difference signal results in more efficient quantization (the signal can be compressed coded with fewer bits).

The G.726 algorithm reduces the bit rate required to transmit intelligible voice, allowing for more channels. The bit rates of 40, 32, 24, and 16 kilobits per second correspond to compression ratios of 1.6:1, 2:1, 2.67:1, and 4:1 with respect to 64 kilobits per second companded PCM. Both G.711 and G.726 are waveform encoders; they can be used to reduce

37367/CAG/B600

the bit rate require to transfer any waveform, like voice, and low bit-rate modem signals, while maintaining an acceptable level of quality.

There exists another class of voice encoders, which model the excitation of the vocal tract to reconstruct a waveform that appears very similar when heard by the human ear, although it may be quite different from the original voice signal. These voice encoders, called vocoders, offer greater voice compression while maintaining good voice quality, at the penalty of higher computational complexity and increased delay.

For the reduction in bit rate over G.711, one pays for an increase in computational complexity. Among voice encoders, the G.726 ADPCM algorithm ranks low to medium on a relative scale of complexity, with companded PCM being of the lowest complexity and code-excited linear prediction (CELP) vocoder algorithms being of the highest.

The G.726 ADPCM algorithm is a sample-based encoder like the G.711 algorithm, therefore, the algorithmic delay is limited to one sample interval. The CELP algorithms operate on blocks of samples (0.625ms to 30 ms for the ITU coder), so the delay they incur is much greater.

The quality of G.726 is best for the two highest bit rates, although it is not as good as that achieved using companded PCM. The quality at 16 kilobits per second is quite poor (a noticeable amount of noise is introduced), and should normally be used only for short periods when it is necessary to conserve network bandwidth (overload situations).

The G.726 interface specifies as input to the G.726 encoder (and output to the G.726 decoder) an 8-bit companded PCM sample according to Recommendation G.711. So strictly speaking, the G.726 algorithm is a transcoder, taking log-PCM and converting it to ADPCM, and vice-versa. Upon input of a companded PCM sample, the G.726 encoder converts it to a 14-bit linear PCM representation for intermediate processing. Similarly, the decoder converts an intermediate 14-bit linear PCM value into an 8-bit companded PCM sample before it is output. An extension of the G.726 algorithm was carried out in 1994 to include, as an option, 14-bit linear PCM input signals and output signals. The specification for such a linear interface is given in Annex A of Recommendation G.726.

37367/CAG/B600

The interface specified by G.726 Annex A bypasses the input and output companded PCM conversions. The effect of removing the companded PCM encoding and decoding is to decrease the coding degradation introduced by the compression and expansion of the linear PCM samples.

The algorithm implemented in the described exemplary embodiment can be the version specified in G.726 Annex A, commonly referred to as G.726A, or any other voice compression algorithm known in the art. Among these voice compression algorithms are those standardized for telephony by the ITU-T. Several of these algorithms operate at a sampling rate of 8000 Hz. with different bit rates for transmitting the encoded voice. By way of example, Recommendations G.729 (1996) and G.723.1 (1996) define code excited linear prediction (CELP) algorithms that provide even lower bit rates than G.711 and G.726. G.729 operates at 8 kbps and G.723.1 operates at either 5.3 kbps or 6.3 kbps.

In an exemplary embodiment, the voice encoder and the voice decoder support one or more voice compression algorithms, including but not limited to, 16 bit PCM (non-standard, and only used for diagnostic purposes); ITU-T standard G.711 at 64 kb/s; G.723.1 at 5.3 kb/s (ACELP) and 6.3 kb/s (MP-MLQ); ITU-T standard G.726 (ADPCM) at 16, 24, 32, and 40 kb/s; ITU-T standard G.727 (Embedded ADPCM) at 16, 24, 32, and 40 kb/s; ITU-T standard G.728 (LD-CELP) at 16 kb/s ; and ITU-T standard G.729 Annex A (CS-ACELP) at 8 kb/s.

The packetization interval for 16 bit PCM, G.711, G.726, G.727 and G.728 should be a multiple of 5 msec in accordance with industry standards. The packetization interval is the time duration of the digital voice samples that are encapsulated into a single voice packet. The voice encoder (decoder) interval is the time duration in which the voice encoder (decoder) is enabled. The packetization interval should be an integer multiple of the voice encoder (decoder) interval (a frame of digital voice samples). By way of example, G.729 encodes frames containing 80 digital voice samples at 8 kHz which is equivalent to a voice encoder (decoder) interval of 10 msec. If two subsequent encoded frames of digital voice sample are collected and transmitted in a single packet, the packetization interval in this case would be 20 msec.

G.711, G.726, and G.727 encodes digital voice samples on a sample by sample basis. Hence, the minimum voice encoder (decoder) interval is 0.125 msec. This is somewhat of a short voice encoder (decoder) interval, especially if the packetization interval is a multiple of 5 msec. Therefore, a single voice packet will contain 40 frames of digital voice samples. G.728 encodes frames containing 5 digital voice samples (or 0.625 msec). A packetization interval of 5 msec

37367/CAG/B600

(40 samples) can be supported by 8 frames of digital voice samples. G.723.1 compresses frames containing 240 digital voice samples. The voice encoder (decoder) interval is 30 msec, and the packetization interval should be a multiple of 30 msec.

Packetization intervals which are not multiples of the voice encoder (or decoder) interval can be supported by a change to the packetization engine or the depacketization engine. This may be acceptable for a voice encoder (or decoder) such as G.711 or 16 bit PCM.

The G.728 standard may be desirable for some applications. G.728 is used fairly extensively in proprietary voice conferencing situations and it is a good trade-off between bandwidth and quality at a rate of 16 kb/s. Its quality is superior to that of G.729 under many conditions, and it has a much lower rate than G.726 or G.727. However, G.728 is MIPS intensive.

Differentiation of various voice encoders (or decoders) may come at a reduced complexity. By way of example, both G.723.1 and G.729 could be modified to reduce complexity, enhance performance, or reduce possible IPR conflicts. Performance may be enhanced by using the voice encoder (or decoder) as an embedded coder. For example, the "core" voice encoder (or decoder) could be G.723.1 operating at 5.3 kb/s with "enhancement" information added to improve the voice quality. The enhancement information may be discarded at the source or at any point in the network, with the quality reverting to that of the "core" voice encoder (or decoder). Embedded coders may be readily implemented since they are based on a given core. Embedded coders are rate scalable, and are well suited for packet based networks. If a higher quality 16 kb/s voice encoder (or decoder) is required, one could use G.723.1 or G.729 Annex A at the core, with an extension to scale the rate up to 16 kb/s (or whatever rate was desired).

The configurable parameters for each voice encoder or decoder include the rate at which it operates (if applicable), which companding scheme to use, the packetization interval, and the core rate if the voice encoder (or decoder) is an embedded coder. For G.727, the configuration is in terms of bits/sample. For example EADPCM(5,2) (Embedded ADPCM, G.727) has a bit rate of 40 kb/s (5 bits/sample) with the core information having a rate of 16 kb/s (2 bits/sample).

6. <u>Packetization Engine</u>

37367/CAG/B600

In an exemplary embodiment, the packetization engine groups voice frames from the voice encoder, and with information from the VAD , creates voice packets in a format appropriate for the packet based network. The two primary voice packet formats are generic voice packets and SID packets. The format of each voice packet is a function of the voice encoder used, the selected packetization interval, and the protocol.

Those skilled in the art will readily recognize that the packetization engine could be implemented in the host. However, this may unnecessarily burden the host with configuration and protocol details, and therefore, if a complete self contained signal processing system is desired, then the packetization engine should be operated in the network VHD. Furthermore, there is significant interaction between the voice encoder, the VAD, and the packetization engine, which further promotes the desirability of operating the packetization engine in the network VHD.

The packetization engine may generate the entire voice packet or just the voice portion of the voice packet. In particular, a fully packetized system with all the protocol headers may be implemented, or alternatively, only the voice portion of the packet will be delivered to the host. By way of example, for VoIP, it is reasonable to create the real-time transport protocol (RTP) encapsulated packet with the packetization engine, but have the remaining transmission control protocol/Internet protocol (TCP/IP) stack residing in the host. In the described exemplary embodiment, the voice packetization functions reside in the packetization engine. The voice packet should be formatted according to the particular standard, although not all headers or all components of the header need to be constructed.

7. Voice Depacketizing Engine / Voice Queue

In an exemplary embodiment, voice de-packetization and queuing is a real time task which queues the voice packets with a time stamp indicating the arrival time. The voice queue should accurately identify packet arrival time within one msec resolution. Resolution should preferably not be less than the encoding interval of the far end voice encoder. The depacketizing engine should have the capability to process voice packets that arrive out of order, and to dynamically switch between voice encoding methods (i.e. between, for example, G.723.1 and G.711). Voice packets should be queued such that it is easy to identify the voice frame to be released, and easy to determine when voice packets have been lost or discarded en route.

37367/CAG/B600

The voice queue may require significant memory to queue the voice packets. By way of example, if G.711 is used, and the worst case delay variation is 250 msec, the voice queue should be capable of storing up to 500 msec of voice frames. At a data rate of 64 kb/s this translates into 4000 bytes or, or 2K (16 bit) words of storage. Similarly, for 16 bit PCM, 500 msec of voice frames require 4K words. Limiting the amount of memory required may limit the worst case delay variation of 16 bit PCM and possibly G.711 This, however, depends on how the voice frames are queued, and whether dynamic memory allocation is used to allocate the memory for the voice frames. Thus, it is preferable to optimize the memory allocation of the voice queue.

The voice queue transforms the voice packets into frames of digital voice samples. If the voice packets are at the fundamental encoding interval of the voice frames, then the delay jitter problem is simplified. In an exemplary embodiment, a double voice queue is used. The double voice queue includes a secondary queue which time stamps and temporarily holds the voice packets, and a primary queue which holds the voice packets, time stamps, and sequence numbers. The voice packets in the secondary queue are disassembled before transmission to the primary queue. The secondary queue stores packets in a format specific to the particular protocol, whereas the primary queue stores the packets in a format which is largely independent of the particular protocol.

In practice, it is often the case that sequence numbers are included with the voice packets, but not the SID packets, or a sequence number on a SID packet is identical to the sequence number of a previously received voice packet. Similarly, SID packets may or may not contain useful information. For these reasons, it may be useful to have a separate queue for received SID packets.

The depacketizing engine is preferably configured to support VoIP, VTOA, VoFR and other proprietary protocols. The voice queue should be memory efficient, while providing the ability to dynamically switch between voice encoders (at the far end), allow efficient reordering of voice packets (used for VoIP) and properly identify lost packets.

8. <u>Voice Synchronization</u>

In an exemplary embodiment, the voice synchronizer analyzes the contents of the voice queue and determines when to release voice frames to the voice decoder, when to play comfort noise, when to perform frame repeats (to cope with lost voice packets or to extend the depth of

37367/CAG/B600

the voice queue), and when to perform frame deletes (in order to decrease the size of the voice queue). The voice synchronizer manages the asynchronous arrival of voice packets. For those embodiments which are not memory limited, a voice queue with sufficient fixed memory to store the largest possible delay variation is used to process voice packets which arrive asynchronously. Such an embodiment includes sequence numbers to identify the relative timings of the voice packets. The voice synchronizer should ensure that the voice frames from the voice queue can be reconstructed into high quality voice, while minimizing the end-to-end delay. These are competing objectives so the voice synchronizer should be configured to provide system trade-off between voice quality and delay.

Preferably, the voice synchronizer is adaptive rather than fixed based upon the worst case delay variation. This is especially true in cases such as VoIP where the worst case delay variation can be on the order of a few seconds. By way of example, consider a VoIP system with a fixed voice synchronizer based on a worst case delay variation of 300 msec. If the actual delay variation is 280 msec, the signal processing system operates as expected. However, if the actual delay variation is 20 msec, then the end -to-end delay is at least 280 msec greater than required. In this case the voice quality should be acceptable, but the delay would be undesirable. On the other hand, if the delay variation is 330 msec then an underflow condition could exist degrading the voice quality of the signal processing system.

The voice synchronizer performs four primary tasks. First, the voice synchronizer determines when to release the first voice frame of a talk spurt from the far end. Subsequent to the release of the first voice frame, the remaining voice frames are released in an isochronous manner. In an exemplary embodiment, the first voice frame is held for a period of time that is equal or less than the estimated worst case jitter.

Second, the voice synchronizer estimates how long the first voice frame of the talk spurt should be held. If the voice synchronizer underestimates the required "target holding time," jitter buffer underflow will likely result. However, jitter buffer underflow could also occur at the end of a talk spurt, or during a short silence interval. Therefore, SID packets and sequence numbers could be used to identify what caused the jitter buffer underflow, and whether the target holding time should be increased. If the voice synchronizer overestimates the required "target holding time," all voice frames will be held too long causing jitter buffer overflow. In response to jitter buffer overflow, the target holding time should be decreased. In the described exemplary embodiment, the voice synchronizer increases the target holding time rapidly for jitter buffer

37367/CAG/B600

underflow due to excessive jitter, but decreases the target holding time slowly when holding times are excessive. This approach allows rapid adjustments for voice quality problems while being more forgiving for excess delays of voice packets.

Thirdly, the voice synchronizer provides a methodology by which frame repeats and frame deletes are performed within the voice decoder. Estimated jitter is only utilized to determine when to release the first frame of a talk spurt. Therefore, changes in the delay variation during the transmission of a long talk spurt must be independently monitored. On buffer underflow (an indication that delay variation is increasing), the voice synchronizer instructs the lost frame recovery engine to issue voice frames repeats. In particular, the frame repeat command instructs the lost frame recovery engine to utilize the parameters from the previous voice frame to estimate the parameters of the current voice frame. Thus, if frames 1, 2 and 3 are normally transmitted and frame 3 arrives late, frame repeat is issued after frame number 2, and if frame number 3 arrives during this period, it is then transmitted. The sequence would be frames 1,2, a frame repeat of frame 2 and then frame 3. Performing frame repeats causes the delay to increase, which increasing the size of the jitter buffer to cope with increasing delay characteristics during long talk spurts. Frame repeats are also issued to replace voice frames that are lost en route.

Conversely, if the holding time is too large due to decreasing delay variation, the speed at which voice frames are released should be increased. Typically, the target holding time can be adjusted, which automatically compresses the following silent interval. However, during a long talk spurt, it may be necessary to decrease the holding time more rapidly to minimize the excessive end to end delay. This can be accomplished by passing two voice frames to the voice decoder in one decoding interval but only one of the voice frames is transferred to the media queue.

The voice synchronizer must also function under conditions of severe buffer overflow, where the physical memory of the signal processing system is insufficient due to excessive delay variation. When subjected to severe buffer overflow, the voice synchronizer could simply discard voice frames.

The voice synchronizer should operate with or without sequence numbers, time stamps, and SID packets. The voice synchronizer should also operate with voice packets arriving out of order and lost voice packets. In addition, the voice synchronizer preferably provides a variety

37367/CAG/B600

of configuration parameters which can be specified by the host for optimum performance, including minimum and maximum target holding time. With these two parameters, it is possible to use a fully adaptive jitter buffer by setting the minimum target holding time to zero msec and the maximum target holding time to 500 msec (or the limit imposed due to memory constraints). Although the preferred voice synchronizer is fully adaptive and able to adapt to varying network conditions, those skilled in the art will appreciate that the voice synchronizer can also be maintained at a fixed holding time by setting the minimum and maximum holding times to be equal.

9. Lost Packet Recovery / Frame Deletion

In applications where voice is transmitted through a packet based network there are instances where not all of the packets reach the intended destination. The voice packets may either arrive too late to be sequenced properly or may be lost entirely. These losses may be caused by network congestion, delays in processing or a shortage of processing cycles. The packet loss can make the voice difficult to understand or annoying to listen to.

Packet recovery refers to methods used to hide the distortions caused by the loss of voice packets. In the described exemplary embodiment, a lost packet recovery engine is implemented whereby missing voice is filled with synthesized voice using the linear predictive coding model of speech. The voice is modelled using the pitch and spectral information from digital voice samples received prior to the lost packets.

The lost packet recovery engine, in accordance with an exemplary embodiment, can be completely contained in the decoder system. The algorithm uses previous digital voice samples or a parametric representation thereof, to estimate the contents of lost packets when they occur.

FIG. 16 shows a block diagram of the voice decoder and the lost packet recovery engine. The lost packet recovery engine includes a voice analyzer 192, a voice synthesizer 194 and a selector 196. During periods of no packet loss, the voice analyzer 192 buffers digital voice samples from the voice decoder 96.

When a packet loss occurs, the voice analyzer 192 generates voice parameters from the buffered digital voice samples. The voice parameters are used by the voice synthesizer 194 to synthesize voice until the voice decoder 96 receives a voice packet, or a timeout period has elapsed. During voice syntheses, a "packet lost" signal is applied to the selector to output the synthesized voice as digital voice samples to the media queue (not shown).

37367/CAG/B600

A flowchart of the lost recovery engine algorithm is shown in FIG. 17A. The algorithm is repeated every frame, whether or not there has been a lost packet. Every time the algorithm is performed, a frame of digital voice samples are output. For purposes of explanation, assume a frame length of 5 ms. In this case, forty samples (5 ms of samples for a sampling rate of 8000 Hz) and a flag specifying whether or not there is voice is buffered in the voice analyzer. The output of the lost recovery engine is also forty digital voice samples.

First, a check is made to see if there has been a packet loss 191. If so, then a check is made to see if this is the first lost packet in a series of voice packets 193. If it is the first lost packet, then the voice is analysed by calculating the LPC parameters, the pitch, and the voicing decision 195 of the buffered digital samples. If the digital samples are voiced 197, then a residual signal is calculated 199 from the buffered digital voice samples and an excitation signal is created from the residual signal 201. The gain factor for the excitation is set to one. If the speech is unvoiced 197, then the excitation gain factor is determined from a prediction error power calculated during a Levinson-Durbin recursion process 207. Using the parameters determined from the voice analysis, one frame of voice is synthesized 201. Finally, the excitation gain factor is attenuated 203, and the synthesized digital voice samples are output 205.

If this is not the first lost packet 193, then a check is made on how many packets have been lost. If the number of lost packets exceeds a threshold 209, then a silence signal is generated and output 211. Otherwise, a frame of digital voice samples are synthesized 201, the excitation gain factor is attenuated 203, and the synthesized digital voice samples are output 205.

If there are decoded digital voice samples 191, then a check is performed to see if there was a lost packet the last time the algorithm was executed 213. If so, then one-half of a frame of digital voice samples are synthesized, and overlap-added with the first one-half of the frame of decoded digital voice samples 215. Then, in all cases, the digital voice samples are buffered in the voice analyser and a frame of digital voice samples is output 217.

a. Calculation of LPC Parameters

There are two main steps in finding the LPC parameters. First the autocorrelation function $r(i)$ is determined up to $r(M)$ where M is the prediction order. Then the Levinson-Durbin recursion formula is applied to the autocorrelation function to get the LPC parameters.

37367/CAG/B600

There are several steps involved in calculating the autocorrelation function. The calculations are performed on the most recent buffered digital voice samples. First, a Hamming window is applied to the buffered samples. Then $r(0)$ is calculated and converted to a floating-point format. Next, $r(1)$ to $r(M)$ are calculated and converted to floating-point. Finally, a conditioning factor is applied to $r(0)$ in order to prevent ill conditioning of the R matrix for a matrix inversion.

The calculation of the autocorrelation function is preferably computationally efficient and makes the best use of fixed point arithmetic. The following equation is used as an estimate of the autocorrelation function from $r(0)$ to $r(M)$:

$$r(i) = \sum_{n=0}^{N-i-1} s[n] \cdot s[n-i]$$

where s[n] is the voice signal and N is the length of the voice window.

The value of $r(0)$ is scaled such that it is represented by a mantissa and an exponent. The calculations are performed using 16 bit multiplications and the summed results are stored in a 40-bit register. The mantissa is found by shifting the result left or right such that the most significant bit is in bit 30 of the 40-bit register (where the least significant bit is bit 0) and then keeping bits 16 to 31. The exponent is the number of left shifts required for normalization of the mantissa. The exponent may be negative if a large amplitude signal is present.

The values calculated for $r(1)$ to $r(M)$ are scaled to use the same exponent as is used for $r(0)$, with the assumption that all values of the autocorrelation function are less than or equal to $r(0)$. This representation in which a series of values are represented with the same exponent is called block floating-point because the whole block of data is represented using the same exponent.

A conditioning factor of 1025/1024 is applied to $r(0)$ in order to prevent ill conditioning of the R matrix. This factor increases the value of $r(0)$ slightly, which has the effect of making $r(0)$ larger than any other value of $r(i)$. It prevents two rows of the R matrix from having equal values or nearly equal values, which would cause ill conditioning of the matrix. When the matrix

37367/CAG/B600

is ill conditioned, it is difficult to control the numerical precision of results during the Levinson-Durbin recursion.

Once the autocorrelation values have been calculated, the Levinson-Durbin recursion formula is applied. In the described exemplary embodiment a sixth to tenth order predictor is preferably used.

Because of truncation effects caused by the use of fixed point calculations, errors can occur in the calculations when the R matrix is ill conditioned. Although the conditioning factor applied to $r(0)$ eliminates this problem for most cases, there is a numerical stability check implemented in the recursion algorithm. If the magnitude of the reflection coefficient gets greater than or equal to one, then the recursion is terminated, the LPC parameters are set to zero, and the prediction error power is set to $r(0)$.

b. Pitch Period and Voicing Calculation.

The voicing determination and pitch period calculation are performed using the zero crossing count and autocorrelation calculations. The two operations are combined such that the pitch period is not calculated if the zero crossing count is high since the digital voice samples are classified as unvoiced. FIG. 17B shows a flowchart of the operations performed.

First the zero crossing count is calculated for a series of digital voice samples 219. The zero crossing count is initialized to zero. The zero crossings are found at a particular point by multiplying the current digital voice sample by the previous digital voice sample and considering the sign of the result. If the sign is negative, then there was a zero crossing and the zero crossing count is incremented. This process is repeated for a number of digital voice samples, and then the zero crossing count is compared to a pre-determined threshold. If the count is above the threshold 221, then the digital voice sample is classified as unvoiced 223. Otherwise, more computations are performed.

Next, if the digital voice samples are not classified as unvoiced, the pitch period is calculated 225. One way to estimate the pitch period in a given segment of speech is to maximize the autocorrelation coefficient over a range of pitch values. This is shown in equation below:

37367/CAG/B600

$$P = \arg\max_p \left(\frac{\sum_{i=0}^{N-p-1} s[i] \cdot s[i+p]}{\sqrt{\sum_{i=0}^{N-p-1} s[i] \cdot s[i]} \cdot \sqrt{\sum_{i=0}^{N-p-1} s[i+p] \cdot s[i+p]}} \right)$$

An approximation to equation the above equation is used to find the pitch period. First the denominator is approximated by $r(0)$ and the summation limit in the numerator is made independent of p as follows

$$P = \arg\max_p \left(\frac{\sum_{i=0}^{N-P_{max}-1} s[i] \cdot s[i+p]}{\sum_{i=0}^{N-P_{max}-1} s[i] \cdot s[i]} \right)$$

where p is the set of integers greater than or equal to P_{min} (preferably on the order of about 20 samples) and less than or equal to P_{max} (preferably on the order of about 130 samples). Next, the denominator is removed since it does not depend on p

$$P = \arg\max_p \left(\sum_{i=0}^{N-P_{max}-1} s[i] \cdot s[i+p] \right)$$

Finally, the speech arrays are indexed such that the most recent samples are emphasized in the estimation of the pitch

37367/CAG/B600

$$P = \operatorname{argmax}_p \left(\sum_{i=0}^{N-P_{max}-1} s[N-1-i] \cdot s[N-1-i-p] \right)$$

This change improves the performance when the pitch is changing in the voice segment under analysis.

When the above equation is applied, a further savings in computations is made by searching only odd values of p. Once the maximum value has been determined, a finer search is implemented by searching the two even values of p on either side of the maximum. Although this search procedure is non-optimal, it normally works well because the autocorrelation function is quite smooth for voiced segments.

Once the pitch period has been calculated, the voicing decision is made using the maximum autocorrelation value 227. If the result is greater than 0.38 times $r(0)$ then the digital samples are classified as voiced 229. Otherwise it is classified as unvoiced 223.

c. Excitation Signal Calculation.

For voiced samples, the excitation signal for voice synthesis is derived by applying the following equation to the buffered digital voice samples:

$$e[n] = s[n] - \sum_{i=1}^{M} a_i \cdot s[n-i]$$

d. Excitation Gain Factor for Unvoiced Speech.

For unvoiced samples, the excitation signal for voice synthesis is a white Gaussian noise sequence with a variance of one quarter. In order to synthesize the voice at the correct level, a gain factor is derived from the prediction error power derived during the Levinson-Durbin recursion algorithm. The prediction error power level gives the power level of the excitation signal that will produce a synthesized voice with power level $r(0)$. Since a gain level is desired rather than a power level, the square root of the prediction error power level is calculated. To make up for the fact that the Gaussian noise has a power of one quarter, the gain is multiplied by a factor of two.

37367/CAG/B600

e. Voiced Synthesis.

The voiced synthesis is performed every time there is a lost voiced packet and also for the first decoded voiced packet after a series of lost packets. FIG. 17C shows the steps performed in the synthesis of voice.

First, the excitation signal is generated. If the samples are voiced 231, then the excitation is generated from the residual signal 233. A residual buffer in the voice analyzer containing the residual signal is modulo addressed such that the excitation signal is equal to repetitions of the past residual signal at the pitch period *P*:

$$e(n) = \begin{cases} e(n-P) & \text{for } n < P \\ e(n-2P) & \text{for } P \le n < 2P \\ e(n-3P) & \text{for } 2P \le n < 3P \\ \ldots & \end{cases}$$

If the value of *P* is less than the number of samples to be synthesized, then the excitation signal is repeated more than once. If *P* is greater than the number of samples to be generated, then less than one pitch period is contained in the excitation. In both cases the algorithm keeps track of the last index into the excitation buffer such that it can begin addressing at the correct point for the next time voice synthesis is required.

If the samples are unvoiced, then a series of Gaussian noise samples are generated 235. Every sample is produced by the addition of twelve uniformly distributed random numbers. Uniformly distributed samples are generated using the linear congruential method (Knuth, 9) as shown by the following equation

$$X_{n+1} = (aX_n + c) \bmod m$$

37367/CAG/B600

where a is set to 32763, c to zero, and m to 65536. The initial value of X_n is equal to 29. The sequence of random numbers repeats every 16384 values, which is the maximum period for the chosen value of m when c is equal to zero. By choosing c not equal to zero the period of repetition could be increased to 65536, but 16384 is sufficient for voice synthesis. The longest segment of voice synthesized by the algorithm is twelve blocks of forty samples, which requires only 5760 uniformly distributed samples. By setting c to zero, the number of operations to calculate the Gaussian random sample is reduced by one quarter.

After the excitation has been constructed, the excitation gain factor is applied to each sample. Finally, the synthesis filter is applied to the excitation to generate the synthetic voice 237.

f. Overlap-Add Calculation.

The overlap-add process is performed when the first good packet arrives after one or more lost packets. The overlap-add reduces the discontinuity between the end of the synthesized voice and the beginning of the decoded voice. To overlap the two voice signals, additional digital voice samples (equal to one-half of a frame) is synthesized and averaged with the first one-half frame of the decoded voice packet. The synthesized voice is multiplied by a down-sloping linear ramp and the decoded voice is multiplied by an up-sloping linear ramp. Then the two signals are added together.

10. DTMF

DTMF (dual-tone, multi-frequency) tones are signaling tones carried within the audio band. A dual tone signal is represented by two sinusoidal signals whose frequencies are separated in bandwidth and which are uncorrelated to avoid false tone detection. A DTMF signal includes one of four tones, each having a frequency in a high frequency band, and one of four tones, each having a frequency in a low frequency band. The frequencies used for DTMF encoding and detection are defined by the ITU and are widely accepted around the world.

In an exemplary embodiment of the present invention, DTMF detection is performed by sampling only a portion of each voice frame. This approach results in improved overall system efficiency by reducing the complexity (MIPS) of the DTMF detection. Although the DTMF is described in the context of a signal processing system for packet voice exchange, those skilled in the art will appreciate that the techniques described for DMTF are likewise suitable for various

37367/CAG/B600

applications requiring signal detection by sampling a portion of the signal. Accordingly, the described exemplary embodiment for DTMF in a signal processing system is by way of example only and not by way of limitation.

There are numerous problems involved with the transmission of DTMF in band over a packet based network. For example, lossy voice compression may distort a valid DTMF tone or sequence into an invalid tone or sequence. Also voice packet losses of digital voice samples may corrupt DTMF sequences and delay variation (jitter) may corrupt the DTMF timing information and lead to lost digits. The severity of the various problems depends on the particular voice decoder, the voice decoder rate, the voice packet loss rate, the delay variation, and the particular implementation of the signal processing system. For applications such as VoIP with potentially significant delay variation, high voice packet loss rates, and low digital voice sample rate (if G.723.1 is used), packet tone exchange is desirable. Packet tone exchange is also desirable for VoFR (FRF-11, class 2). Thus, proper detection and out of band transfer via the packet based network is useful.

The ITU and Bellcore have promulgated various standards for DTMF detectors. The described exemplary DTMF detector preferably complies with ITU-T Standard Q.24 (for DTMF digit reception) and Bellcore GR-506-Core, TR-TSY-000181, TR-TSY-000762 and TR-TSY-000763, the contents of which are hereby incorporated by reference as though set forth in full herein. These standards involve various criteria, such as frequency distortion allowance, twist allowance, noise immunity, guard time, talk-down, talk-off, acceptable signal to noise ratio, and dynamic range, etc. which are summarized in the table below.

The distortion allowance criteria specifies that a DTMF detector should detect a transmitted signal that has a frequency distortion of less than 1.5% and should not detect any DTMF signals that have frequency distortion of more than 3.5%. The term "twist" refers to the difference, in decibels, between the amplitude of the strongest key pad column tone and the amplitude of the strongest key pad row tone. For example, the Bellcore standard requires the twist to be between -8 and +4 dBm. The noise immunity criteria requires that if the signal has a signal to noise ratio (SNR) greater than certain decibels, then the DTMF detector is required to not miss the signal, i.e., is required to detect the signal. Different standards have different SNR requirements, which usually range from 12 to 24 decibels. The guard time check criteria requires that if a tone has a duration greater than 40 milliseconds, the DTMF detector is required to detect the tone, whereas if the tone has a duration less than 23 milliseconds, the DTMF

37367/CAG/B600

detector is required to not detect the tone. Similarly, the DTMF detector is required to accept interdigit intervals which are greater than or equal to 40 milliseconds. Alternate embodiments of the present invention readily provide for compliance with other telecommunication standards such as EIA-464B, and JJ-20.12.

Referring to FIG. 18 the DTMF detector 76 processes the 64kb/s pulse code modulated (PCM) signal, i.e., digital voice samples 76(a) buffered in the media queue (not shown). The input to the DTMF detector 76 should preferably be sampled at a rate that is at least higher than approximately 4 kHz or twice the highest frequency of a DTMF tone. If the incoming signal (i.e., digital voice samples) is sampled at a rate that is greater than 4 kHz (i.e. Nyquist for highest frequency DTMF tone) the signal may immediately be downsampled so as to reduce the complexity of subsequent processing. The signal may be downsampled by filtering and discarding samples.

A block diagram of an exemplary embodiment of the invention is shown in FIG. 18. The described exemplary embodiment includes a system for processing the upper frequency band tones and a substantially similar system for processing the lower frequency band tones. A filter 210 and sampler 212 may be used to down-sample the incoming signal. In the described exemplary embodiment, the sampling rate is 8 kHz and the front end filter 210 and sampler 212 do not down-sample the incoming signal. The output of the sampler 212 is filtered by two bandpass filters $H_h(z)$ 214 and $G_h(z)$ 216 for the upper frequency band and $H_l(z)$ 218 and $G_l(Z)$ 220 for the lower frequency band) and down-sampled by samplers 222,224 for the upper frequency band and 226,228 for the lower frequency band. The bandpass filters (214, 216 and 218,220) for each frequency band are designed using a pair of lowpass filters, one filter H(z) which multiplies the down-sampled signal by $\cos(2\pi f_h nT)$ and the other filter G(z) which multiplies the down-sampled signal by $\sin(2\pi f_h nT)$ (where $T = 1/f_s$ where f_s is the sampling frequency after the front end down-sampling by the filter 210 and the sampler 212.

In the described exemplary embodiment, the bandpass filters (214, 216 and 218,220) are executed every eight samples and the outputs (214a, 216a and 218a, 220a) of the bandpass filters (214, 216 and 218,220) are down-sampled by samplers 222, 224 and 226, 228 at a ratio of eight to one. The combination of down-sampling is selected so as to optimize the performance of a particular DSP in use and preferably provides a sample approximately every msec or a 1 kbs signal. Down-sampled signals in the upper and lower frequency bands respectively are real signals. In the upper frequency band, a multiplier 230 multiplies the output of sampler 224 by the square root of minus one (i.e. j) 232. A summer 234 then adds the output of downsampler 222 with the imaginary signal 230(a). Similarly, in the lower frequency band, a multiplier 236 multiplies the output of downsampler 228 by the square root of minus one (i.e. j) 238. A summer

37367/CAG/B600

240 then adds the output of downsampler 226 with the imaginary signal 236(a). Combined signals $x_h(t)$ 234(a) and $x_l(t)$ 240(a) at the output of the summers 234, 240 are complex signals. It will be appreciated by one of skill in the art that the function of the bandpass filters can be accomplished by alternative finite impulse response filters or structures such as windowing followed by DFT processing.

If a single frequency is present within the bands defined by the bandpass filters, the combined complex signals $x_h(t)$ and $x_l(t)$ will be constant envelope (complex) signals. Short term power estimator 242 and 244 measure the power of $x_h(t)$ and $x_l(t)$ respectively and compare the estimated power levels of $x_h(t)$ and $x_l(t)$ with the requirements promulgated in ITU-T Q.24. In the described exemplary embodiment, the upper band processing is first executed to determine if the power level within the upper band complies with the thresholds set forth in the ITU-T Q.24 recommendations. If the power within the upper band does not comply with the ITU-T recommendations the signal is not a DTMF tone and processing is terminated. If the power within the upper band complies with the ITU-T Q.24 standard, the lower band is processed. A twist estimator 246 compares the power in the upper band and the lower band to determine if the twist (defined as the ratio of the power in the lower band and the power in the upper band) is within an acceptable range as defined by the ITU-T recommendations. If the ratio of the power within the upper band and lower band is not within the bounds defined by the standards, a DTMF tone is not present and processing is terminated.

If the ratio of the power within the upper band and lower band complies with the thresholds defined by the ITU-T Q.24 and Bellcore GR-506-Core, TR-TSY-000181, TR-TSY-000762 and TR-TSY-000763 standards, the frequency of the upper band signal $x_h(t)$ and the frequency of the lower band signal $x_l(t)$ are estimated. Because of the duration of the input signal (one msec), conventional frequency estimation techniques such as counting zero crossings may not sufficiently resolve the input frequency. Therefore, differential detectors 248 and 250 are used to estimate the frequency of the upper band signal $x_h(t)$ and the lower band signal $x_l(t)$ respectively. The differential detectors 248 and 250 estimate the phase variation of the input signal over a given time range. Advantageously, the accuracy of estimation is substantially insensitive to the period over which the estimation is performed. With respect to upper band input $x_h(n)$, (and assuming $x_h(n)$ is a sinusoid of frequency f_i) the differential detector 248 computes:

$$y_h(n) = x_h(n)x_h(n-1)^*e(-j2\pi f_{mid})$$

where f_{mid} is the mean of the frequencies in the upper band or lower band and superscript* implies complex conjugation. Then,

37367/CAG/B600

$$y_h(n) = e(j2\pi f_i n)\, e(-j2\pi f_i(n-1)) e(-j2\pi f_{mid}) = e(j2\pi(f_i - f_{mid}))$$

which is a constant, independent of n. Arctan functions 252 and 254 each takes the complex input and computes the angle of the above complex value that uniquely identifies the frequency present in the upper and lower bands. In operation $atan2(\sin(2\pi(f_i-f_{mid})), \cos(2\pi(f_i-f_{mid})))$ returns to within a scaling factor the frequency difference f_i-f_{mid}. Those skilled in the art will appreciate that various algorithms, such as a frequency discriminator, could be use to estimate the frequency of the DTMF tone by calculating the phase variation of the input signal over a given time period.

Having estimated the frequency components of the upper band and lower band, the DTMF detector analyzes the upper band and lower band signals to determine whether a DTMF digit is present in the incoming signals and if so which digit. Frequency calculators 256 and 258 compute a mean and variance of the frequency deviation over the entire window of frequency estimates to identify valid DTMF tones in the presence of background noise or speech that resembles a DTMF tone. In the described exemplary embodiment, if the mean of the frequency estimates over the window is within acceptable limits, preferably less than +/-2.8% for the lowband and +/-2.5% for the highband the variance is computed. If the variance is less than a predetermined threshold, preferably on the order of about 1464 Hz^2 (i.e. standard deviation of 38.2 Hz) the frequency is declared valid. Referring to FIG. 18A, DTMF control logic 259 compares the frequency identified for the upper and lower bands to the frequency pairs identified in the ITU-T recommendations to identify the digit. The DTMF control logic 259 forwards a tone detection flag 259(b) to a state machine 260. The state machine 260 analyzes the time sequence of events and compares the tone on and tone off periods for a given tone to the ITU-T recommendations to determine whether a valid dual tone is present. In the described exemplary embodiment the total window size is preferably 5 msec so that a DTMF detection decision is performed every 5 msec.

In the context of an exemplary embodiment of the voice mode, the DTMF detector is operating in the packet tone exchange along with a voice encoder operating under the packet voice exchange, which allows for simplification of DTMF detection processing. Most voice encoders operate at a particular frame size (the number of voice samples or time in msec over which voice is compressed). For example, the frame size for ITU-T standard G.723.1 is 30 msec. For ITU-T standard G.729 the frame size is 10 msec. In addition, many packet voice systems group multiple output frames from a particular voice encoder into a network cell or packet. To prevent leakage through the voice path, the described exemplary embodiment delays DTMF detection until the last frame of speech is processed before a full packet is constructed. Therefore, for transmissions in accordance with the G.723.1 standard and a single output frame

37367/CAG/B600

placed into a packet, DTMF detection may be invoked every 30 msec (synchronous with the end of the frame). Under the G.729 standard with two voice encoder frames placed into a single packet, DTMF detection or decision may be delayed until the end of the second voice frame within a packet is processed.

In the described exemplary embodiment, the DTMF detector is inherently stateless, so that detection of DTMF tones within the second 5 msec DTMF block of a voice encoder frame doesn't depend on DTMF detector processing of the first 5 msec block of that frame. If the delay in DTMF detection is greater than or equal to twice the DTMF detector block size, the processing required for DTMF detection can be further simplified. For example, the instructions required to perform DTMF detection may be reduced by 50% for a voice encoder frame size of 10 msec and a DTMF detector frame size of 5 msec. The ITU-T Q.24 standard requires DTMF tones to have a minimum duration of 23 msec and an inter-digit interval of 40 msec. Therefore, by way of example, a valid DTMF tone may be detected within a given 10 msec frame by only analyzing the second 5 msec interval of that frame. Referring to FIG. 18A, in the described exemplary embodiment, the DTMF control logic 259 analyzes DTMF detector output 76(a) and selectively enables DTMF detection analysis 259(a) for a current frame segment, as a function of whether a valid dual tone was detected in previous and future frame segments. For example, if a DTMF tone was not detected in the previous frame and if DTMF is not present in the second 5 msec interval of the current frame, then the first 5 msec block need not be processed so that DTMF detection processing is reduced by 50%. Similar savings may be realized if the previous frame did contain a DTMF (if the DTMF is still present in the second 5 msec portion it is most likely it was on in the first 5 msec portion). This method is easily extended to the case of longer delays (30 msec for G.723.1 or 20-40 msec for G.729 and packetization intervals from 2-4 or more). It may be necessary to search more than one 5 msec period out of the longer interval, but only a subset is necessary.

DTMF events are preferably reported to the host. This allows the host, for example, to convert the DTMF sequence of keys to a destination address. It will, therefore, allow the host to support call routing via DTMF.

Depending on the protocol, the packet tone exchange may support muting of the received digital voice samples, or discarding voice frames when DTMF is detected. In addition, to avoid DTMF leakage into the voice path, the voice packets may be queued (but not released) in the encoder system when DTMF is pre-detected. DTMF is pre-detected through a combination of DTMF decisions and state machine processing. The DTMF detector will make a decision (i.e. is there DTMF present) every five msec. A state machine 260 analyzes the history of a given DTMF tone to determine the current duration of a given tone so as to estimate how long the tone

will likely continue. If the detection was false (invalid), the voice packets are ultimately released, otherwise they are discarded. This will manifest itself as occasional jitter when DTMF is falsely pre-detected. It will be appreciated by one of skill in the art that tone packetization can alternatively be accomplished through compliance with various industry standards such as for example, the Frame Relay Forum (FRF-11) standard, the voice over atm standard ITU-T I.363.2, and IETF-draft-avt-tone-04, RTP Payload for DTMF Digits for Telephony Tones and Telephony Signals, the contents of which are hereby incorporated by reference as though set forth in full.

Software to route calls via DTMF can be resident on the host or within the signal processing system. Essentially, the packet tone exchange traps DTMF tones and reports them to the host or a higher layer. In an exemplary embodiment, the packet tone exchange will generate dial tone when an off-hook condition is detected. Once a DTMF digit is detected, the dial tone is terminated. The packet tone exchange may also have to play ringing tone back to the near end user (when the far end phone is being rung), and a busy tone if the far end phone is unavailable. Other tones may also need to be supported to indicate all circuits are busy, or an invalid sequence of DTMF digits were entered.

11. Call progress tone Detection

Telephone systems provide users with feedback about what they are doing in order to simplify operation and reduce calling errors. This information can be in the form of lights, displays, or ringing, but is most often audible tones heard on the phone line. These tones are generally referred to as call progress tones, as they indicate what is happening to dialed phone calls. Conditions like busy line, ringing called party, bad number, and others each have distinctive tone frequencies and cadences assigned them for which some standards have been established. A call progress tone signal includes one of four tones. The frequencies used for call progress tone encoding and detection, namely 350, 440, 480, and 620 Hz, are defined by the international telecommunication union and are widely accepted around the world. The relatively narrow frequency separation between tones, 40Hz in one instance complicates the detection of individual tones. In addition, the duration or cadence of a given tone is used to identify alternate conditions.

An exemplary embodiment of the call progress tone detector analyzes the spectral (frequency) characteristics of an incoming telephony voice-band signal and generates a tone detection flag as a function of the spectral analysis. The temporal (time) characteristics of the tone detection flags are then analyzed to detect call progress tone signals. The call progress tone detector then forwards the call progress tone signal to the packetization engine to be packetized and transmitted across the packet based network. Although the call progress tone detector is

37367/CAG/B600

described in the context of a signal processing system for packet voice exchange, those skilled in the art will appreciate that the techniques described for call progress tone detection are likewise suitable for various applications requiring signal detection by analyzing spectral or temporal characteristics of the signal. Accordingly, the described exemplary embodiment for precision tone detection in a signal processing system is by way of example only and not by way of limitation.

The described exemplary embodiment preferably includes a call progress tone detector that operates in accordance with industry standards for the power level (Bellcore SR3004-CPE Testing Guidelines; Type III Testing) and cadence (Bellcore GR506-Core and Bellcore LSSGR Signaling For Analog Interface, Call Purpose Signals) of a call progress tone. The call progress tone detector interfaces with the media queue to detect incoming call progress tone signals such as dial tone, re-order tone, audible ringing and line busy or hook status. The problem of call progress tone signaling and detection is a common telephony problem. In the context of packet voice systems in accordance with an exemplary embodiment of the present invention, telephony devices are coupled to a signal processing system which, for the purposes of explanation, is operating in a network gateway to support the exchange of voice between a traditional circuit switched network and a packet based network. In addition, the signal processing system operating on network gateways also supports the exchange of voice between the packet based network and a number of telephony devices.

Referring to FIG. 19 the call progress tone detector 264 continuously monitors the media queue 66 of the voice encoder system. Typically the call progress tone detector 264 is invoked every ten msec. Thus, for an incoming signal sampled at a rate of 8 kHz, the preferred call progress tone detector operates on blocks of eighty samples. The call progress tone detector 264 includes a signal processor 266 which analyzes the spectral characteristics of the samples buffered in the media queue 66. The signal processor 266 performs anti-aliasing, decimation, bandpass filtering, and frequency calculations to determine if a tone at a given frequency is present. A cadence processor 268 analyzes the temporal characteristics of the processed tones by computing the on and off periods of the incoming signal. If the cadence processor 268 detects a call progress tone for an acceptable on and off period in accordance with the Bellcore GR506-Core standard, a "Tone Detection Event" will be generated.

A block diagram for an exemplary embodiment of the signal processor 266 is shown in FIG. 20. An anti-aliasing low pass filter 270, with a cutoff frequency of preferably about 666Hz, filters the samples buffered in the media queue so as to remove frequency components above the highest call progress tone frequency, i.e. 660 Hz. A down sampler 272 is coupled to the output of the low pass filter 270. Assuming an 8 kHz input signal, the down sampler 272 preferably

37367/CAG/B600

decimates the low pass filtered signal at a ratio of six:one (which avoids aliasing due to under sampling). The output 272(a) of down sampler 272 is filtered by eight bandpass filters (274, 276, 278, 280, 282, 284, 286 and 288), (i.e. two filters for each call progress tone frequency). The decimation effectively increases the separation between tones, so as to relax the roll-off requirements (i.e. reduce the number of filter coefficients) of the bandpass filters 274-288 which simplifies the identification of individual tones. In the described exemplary embodiment, the bandpass filters for each call progress tone 274-288 are designed using a pair of lowpass filters, one filter which multiplies the down sampled signal by $\cos(2\pi f_h nT)$ and the other filter which multiplies the down sampled signal by $\sin(2\pi f_h nT)$ (where $T = 1/f_s$ where f_s is the sampling frequency after the decimation by the down sampler 272. The outputs of the band pass filters are real signals. Multipliers (290, 292, 294 and 296) multiply the outputs of filters (276, 280, 284 and 288) respectively by the square root of minus one (i.e. j) 298 to generate an imaginary component. Summers (300, 302, 304 and 306) then add the outputs of filters (274, 278, 282 and 286) with the imaginary components (290a, 292a, 294a and 296a) respectively. The combined signals are complex signals. It will be appreciated by one of skill in the art that the function of the bandpass filters (274-288) can be accomplished by alternative finite impulse response filters or structures such as windowing followed by DFT processing.

Power estimators (308, 310, 312 and 314) estimate the short term average power of the combined complex signals (300a, 302a, 304a and 306a) for comparison to power thresholds determined in accordance with the recommended standard (Bellcore SR3004-CPE Testing Guidelines For Type III Testing). The power estimators 308-312 forward an indication to power state machines (316, 318, 320 and 322) respectively which monitor the estimated power levels within each of the call progress tone frequency bands. Referring to FIG. 21, the power state machine is a three state device, including a disarm state 324, an arm state 326, and a power on state 328. As is known in the art, the state of a power state machine depends on the previous state and the new input. For example, if an incoming signal is initially silent, the power estimator 308 would forward an indication to the power state machine 316 that the power level is less than the predetermined threshold. The power state machine would be off, and disarmed. If the power estimator 308 next detects an incoming signal whose power level is greater than the predetermined threshold, the power estimator forwards an indication to the power state machine 316 indicating that the power level is greater than the predetermined threshold for the given incoming signal. The power state machine 316 switches to the off but armed state. If the next input is again above the predetermined threshold, the power estimator 308 forwards an indication to the power state machine 316 indicating that the power level is greater than the predetermined threshold for the given incoming signal. The power state machine 316 now toggles to the on and armed state. The power state machine 316 substantially reduces or eliminates false detections due to glitches, white noise or other signal anomalies.

37367/CAG/B600

Turning back to FIG. 20, when the power state machine is set to the on state, frequency calculators (330, 332, 334 and 336) estimate the frequency of the combined complex signals. The frequency calculators (330-336), utilize a differential detection algorithm to estimate the frequency within each of the four call progress tone bands. The frequency calculators (330-336) estimate the phase variation of the input signal over a given time range. Advantageously, the accuracy of the estimation is substantially insensitive to the period over which the estimation is performed. Assuming a sinusoidal input x(n) of frequency f_i the frequency calculator computes:

$$y(n) = x(n)x(n-1)^{*}e(-j2\pi f_{mid})$$

where f_{mid} is the mean of the frequencies within the given call progress tone group and superscript* implies complex conjugation. Then,

$$y(n) = e(j2\pi f_i n)\ e(-j2\pi f_i(n-1))e(-j2\pi f_{mid})$$

$$= e(j2\pi(f_i - f_{mid}))$$

which is a constant, independent of n. The frequency calculators (330-336) then invoke an arctan function that takes the complex signal and computes the angle of the above complex value that identifies the frequency present within the given call progress tone band. In operation $\operatorname{atan2}(\sin(2\pi(f_i-f_{mid})), \cos(2\pi(f_i-f_{mid})))$ returns to within a scaling factor the frequency difference f_i-f_{mid}. Those skilled in the art will appreciate that various algorithms, such as a frequency discriminator, could be use to estimate the frequency of the call progress tone by calculating the phase variation of the input signal over a given time period.

The frequency calculators (330-336) compute the mean of the frequency deviation over the entire 10 msec window of frequency estimates to identify valid call progress tones in the presence of background noise or speech that resembles a call progress tone. If the mean of the frequency estimates over the window is within acceptable limits as summarized by the table below, a tone on flag is forwarded to the cadence processor. The frequency calculators (330-336) are preferably only invoked if the power state machine is in the on state thereby reducing the processor loading (i.e. fewer MIPS) when a call progress tone signal is not present.

Tone	Frequency One / Mean	Frequency Two / Mean
Dial Tone	350 Hz / 2 Hz	440 Hz / 3 Hz
Busy	480 Hz / 7 Hz	620 Hz / 9 Hz
Re-order	480 Hz / 7 Hz	620 Hz / 9 Hz

37367/CAG/B600

Audible Ringing	440 Hz / 7 Hz	480 Hz / 7 Hz

Referring to FIG. 22A, the signal processor 266 forwards a tone on / tone off indication to the cadence processor 268 which considers the time sequence of events to determine whether a call progress tone is present. Referring to FIG. 22, in the described exemplary embodiment, the cadence processor 268 preferably comprises a four state, cadence state machine 340, including a cadence tone off state 342, a cadence tone on state 344, a cadence tone arm state 346 and an idle state 348 (see FIG. 22). The state of the cadence state machine 340 depends on the previous state and the new input. For example, if an incoming signal is initially silent, the signal processor would forward a tone off indication to the cadence state machine 340. The cadence state machine 340 would be set to a cadence tone off and disarmed state. If the signal processor next detects a valid tone, the signal processor forwards a tone on indication to the cadence state machine 340. The cadence state machine 340 switches to a cadence off but armed state. Referring to FIG. 22A, the cadence state machine 340 preferably invokes a counter 350 that monitors the duration of the tone indication. If the next input is again a valid call progress tone, the signal processor forwards a tone on indication to the cadence state machine 340. The cadence state machine 340 now toggles to the cadence tone on and cadence tone armed state. The cadence state machine 340 would remain in the cadence tone on state until receiving two consecutive tone off indications from the signal processor at which time the cadence state machine 340 sends a tone off indication to the counter 350. The counter 350, resets and forwards the duration of the on tone to cadence logic 352. The cadence processor 268 similarly estimates the duration of the off tone, which the cadence logic 352 utilizes to determine whether a particular tone is present by comparing the duration of the on tone, off tone signal pair at a given tone frequency to the tone plan recommended in industry standard as summarized in the table below.

Tone	Duration of Tone On / Tolerance	Duration of Tone Off / Tolerance
Dial Tone	Continuous On	No Off Tone
Busy	500 msec / (+/-50 msec)	500 msec / (+/-50 msec)
Re-order	250 msec / (+/-25 msec)	200 msec / (+/-25 msec)
Audible Ringing	1000 msec / (+/-200 msec)	3000 msec / (+/-2000 msec)
Audible Ringing (Tone 2)	2000 msec / (+/-200 msec)	4000 msec / (+/-2000 msec)

37367/CAG/B600

12. Resource Manager

In the described exemplary embodiment utilizing a multi-layer software architecture operating on a DSP platform, the DSP server includes networks VHDs (see FIG. 7). Each network VHD can be a complete self-contained software module for processing a single channel with a number of different telephony devices. Multiple channel capability can be achieved by adding network VHDs to the DSP server. The resource manager dynamically controls the creation and deletion of VHDs and services.

In the case of multi-channel communications using a number of network VHDs, the services invoked by the network VHDs and the associated PXDs are preferably optimized to minimize system resource requirements in terms of memory and/or computational complexity. This can be accomplished with the resource manager which reduces the complexity of certain algorithms in the network VHDs based on predetermined criteria. Although the resource management processor is described in the context of a signal processing system for packet voice exchange, those skilled in the art will appreciate that the techniques described for resource management processing are likewise suitable for various applications requiring processor complexity reductions. Accordingly, the described exemplary embodiment for resource management processing in a signal processing system is by way of example only and not by way of limitation.

In one embodiment, the resource manager can be implemented to reduce complexity when the worst case system loading exceeds the peak system resources. The worst case system loading is simply the sum of the worst case (peak) loading of each service invoked by the network VHD and its associated PXDs. However, the statistical nature of the processor resources required to process voice band telephony signals is such that it is extremely unlikely that the worst case processor loading for each PXD and /or service will occur simultaneously. Thus, a more robust (lower overall power consumption and higher densities, i.e. more channels per DSP) signal processing system may be realized if the average complexity of the various voice mode PXDs and associated services is minimized. Therefore, in the described exemplary embodiment, average system complexity is reduced and system resources may be over subscribed (peak loading exceeds peak system resources) in the short term wherein complexity reductions are invoked to reduce the peak loading placed on the system.

The described exemplary resource manager should preferably manage the internal and external program and data memory of the DSP. The transmission / signal processing of voice is inherently dynamic, so that the system resources required for various stages of a conversation are time varying. The resource manager should monitor DSP resource utilization and dynamically

37367/CAG/B600

allocate resources to numerous VHDs and PXDs to achieve a memory and computationally (reduced MIPS) efficient system. For example, when the near end talker is actively speaking, the voice encoder consumes significant resources, but the far end is probably silent so that the echo canceller is probably not adapting and may not be executing the transversal filter. When the far end is active, the near end is most likely inactive, which implies the echo canceller is both canceling far end echo and adapting. However, when the far end is active the near end is probably inactive, which implies that the VAD is probably detecting silence and the voice encoder consumes minimal system resources. Thus, it is unlikely that the voice encoder and echo canceller resource utilization peak simultaneously. Furthermore, if processor resources are taxed, echo canceller adaptation may be disabled if the echo canceller is adequately adapted or interleaved (adaptation enabled on alternating echo canceller blocks) to reduce the computational burden placed on the processor.

Referring to FIG. 23, in the described exemplary embodiment, the resource manager 351 manages the resources of two network VHDs 62', 62" and their associated PXDs 60', 60". Initially, the average complexity of the services running in each VHD and its associated PXD is reported to the resource manager. The resource manager 351 sums the reported complexities to determine whether the sum exceeds the system resources. If the sum of the average complexities reported to the resource manager 351 are within the capability of the system resources, no complexity reductions are invoked by the resource manager 351. Conversely, if the sum of the average complexities of the services running in each VHD and its associated PXD overload the system resources, then the resource manager can invoke a number of complexity reduction methodologies. For example, the echo cancellers 70', 70" can be forced into the bypass mode (see FIG. 11) and/or the echo canceller adaption can be reduced or disabled. In addition (or in the alternative), complexity reductions in the voice encoders 82', 82" and voice decoders 96', 96" can be invoked.

The described exemplary embodiment may reduce the complexity of certain voice mode services and associated PXDs so as to reduce the computational / memory requirements placed upon the system. Various modifications to the voice encoders may be included to reduce the load placed upon the system resources. For example, the complexity of a G.723.1 voice encoder may be reduced by disabling the post filter in accordance with the ITU-T G.723.1 standard which is incorporated herein by reference as if set forth in full. Also the voicing decision may be modified so as to be based on the open loop normalized pitch correlation computed at the open loop pitch lag L determined by the standard voice encoding algorithm. This entails a modification to the ITU-T G.723.1 C language routine Estim_Pitch(). If d(n) is the input to the pitch estimation function, the normalized open loop pitch correlation at the open loop pitch lag L is:

37367/CAG/B600

$$X(L)=\frac{\left(\sum_{n=0}^{N-1}(d(n)(dn-L))\right)^2}{\left(\sum_{n=o}^{N-1}d(n)^2 \times \sum_{n=0}^{N-1}d(n-L)^2\right)}$$

where N is equal to a duration of 2 subframes (or 120 samples).

Also, the ability to bypass the adaptive codebook based on a threshold computed from a combination of the open loop normalized pitch correlation and speech/residual energy may be included. In the standard encoder, the search through the adaptive codebook gain codebook begins at index zero and may be terminated before the entire codebook is searched (less than the total size of the adaptive codebook gain codebook which is either 85 or 170 entries) depending on the accumulation of potential error. A preferred complexity reduction truncates the adaptive codebook gain search procedure if the open loop normalized pitch correlation and speech/residual energy meets a certain by searching entries from:

- the upper bound (computed in the standard coder) less half the adaptive codebook size (or index zero, whichever is greater) for voiced speech; and

- from index zero up to half the size of the adaptive code gain codebook (85/2 or 170/2).

The adaptive codebook may also be completely bypassed under some conditions by setting the adaptive codebook gain index to zero, which selects an all zero adaptive codebook gain setting.

The fixed excitation in the standard encoder may have a periodic component. In the standard encoder, if the open loop pitch lag is less than the subframe length minus two, then a excitation search function (the function call Find_Best() in the ITU-T G.723.1 C language simulation) is invoked twice. To reduce system complexity, the fixed excitation search procedure may be modified (at 6.3 kb/s) such that the fixed excitation search function is invoked once per invocation of the fixed excitation search procedure (routine Find_Fcbk()). If the open loop pitch lag is less than the subframe length minus two then a periodic repetition is forced, otherwise there is no periodic repetition (as per the standard encoder for that range of open loop pitch lags). In the described complexity reduction modification, the decision on which manner to invoke it is based on the open loop pitch lag and the voicing strength.

Similarly, the fixed excitation search procedure can be modified (at 5.3 kb/s) such that a higher threshold is chosen for voice decisions. In the standard encoder, the voicing decision is considered to be voiced of the open loop normalized pitch correlation is greater than 0.5 (variable named "threshold" in the ITU-T G.723.1) is set to 0.5. In a modification to reduce the

complexity of this function, the threshold may be set to 0.75. This greatly reduces the complexity of the excitation search procedure while avoiding substantial impairment to the voice quality.

Similar modifications may be made to reduce the complexity of a G.729 Annex A voice encoder. For example, the complexity of a G.729 Annex A voice encoder may be reduced by disabling the post filter in accordance with the G.729 Annex A standard which is incorporated herein by reference as if set out in full. Also, the complexity of a G.729 Annex A voice encoder may be further reduced by including the ability to bypass the adaptive codebook or reduce the complexity of the adaptive codebook search significantly. In the standard voice encoder, the adaptive codebook searches over a range of lags based on the open loop pitch lag. The adaptive codebook bypass simply chooses the minimum lag. The complexity of the adaptive codebook search may be reduced by truncating the adaptive codebook search such that fractional pitch periods are not considered within the search (not searching the non-integer lags). These modifications are made to the ITU-T G.729 Annex A, C language routine Pitch_fr3_fast(). The complexity of a G.729 Annex A voice encoder may be further reduced by substantially reducing the complexity of the fixed excitation search. The search complexity may be reduced by bypassing the depth first search 4, phase A: track 3 and 0 search and the depth first search 4, phase B: track 1 and 2 search.

Each modification reduces the computational complexity but also minimally reduces the resultant voice quality. However, since the voice encoders are externally managed by the resource manager to minimize occasional system resource overloads, the voice encoder should predominately operate with no complexity reductions. The preferred embedded software embodiment should include the standard code as well as the modifications required to reduce the system complexity. The resource manager should preferably minimize power consumption and computational cycles by invoking complexity reductions which have substantially no impact on voice quality. The different complexity reductions schemes should be selected dynamically based on the processing requirements for the current frame (over all voice channels) and the statistics of the voice signals on each channel (voice level, voicing, etc).

Although complexity reductions are rare, the appropriate PXDs and associated services invoked in the network VHDs should preferably incorporate numerous functional features to accommodate such complexity reductions. For example, the appropriate voice mode PXDs and associated services should preferably include a main routine which executes the complexity reductions described above with a variety of complexity levels. For example, various complexity levels may be mandated by setting various complexity reduction flags. In addition, the resource manager should accurately measure the resource requirements of PXDs and services with fixed resource requirements (i.e. complexity is not controllable), to support the computation of peak

complexity and average complexity. Also, a function that returns the estimated complexity in cycles according to the desired complexity reduction level should preferably be included.

The described exemplary embodiment preferably includes four complexity reduction levels. In the first level, all complexity reductions are disabled so that the complexity of the PXDs and services is not reduced.

The second level provides minimal or transparent complexity reductions (reductions which should preferably have substantially no observable impact on performance under most conditions). In the transparent mode the voice encoders (G.729, G.723.1) preferably use voluntary reductions and the echo canceller is forced into the bypass mode and adaption is toggled (i.e., adaptive is enabled for every other frame). Voluntary reductions for G.723.1 voice encoders are preferably selected as follows. First, if the frame energy is less than -55 dBm0, then the adaptive codebook is bypassed and the fixed excitation searches are reduced, as per above. If the frame energy is less than -45 dBm0 but greater than -55 dBm0, then the adaptive codebook is partially searched and the fixed excitation searches are reduced as per above. In addition, if the open loop normalized pitch correlation is less than 0.305 then the adaptive codebook is partially searched. Otherwise, no complexity reductions are done. Similarly, voluntary reductions for the G.729 voice encoders preferably proceed as follows: first, if the frame energy is less than -55 dBm0, then the adaptive codebook is bypassed and the fixed excitation search is reduced per above. Next if the frame energy is less than -45 dBm0 but greater than -55 dBm0, then the reduced complexity adaptive codebook is used and the excitation search complexity is reduced. Otherwise, no complexity reduction is used.

The third level of complexity reductions provides minor complexity reductions (reductions which may result in a slight degradation of voice quality or performance). For example, in the third level the voice encoders preferably use voluntary reductions, "find_best" reduction (G.723.1), fixed codebook threshold change (5.3 kbps G.723.1), open loop pitch search reduction (G.723.1 only), and minimal adaptive codebook reduction (G.729 and G.723.1). In addition, the echo canceller is forced into the bypass mode and adaption is toggled.

In the fourth level major complexity reductions occur, that is reductions which should noticeably effect the performance quality. For example, in the fourth level of complexity reductions the voice encoders use the same complexity reductions as those used for level three reductions, as well as adding a bypass adaptive codebook reduction (G.729 and G.723.1). In addition, the echo canceller is forced into the bypass mode and adaption is completely disabled. The resource manager preferably limits the invocation of fourth level major reductions to extreme circumstances, such as, for example when there is double talk on all active channels.

37367/CAG/B600

The described exemplary resource manager monitors system resource utilization. Under normal system operating conditions, complexity reductions are not mandated on the echo canceller or voice encoders. Voice/FAX and data traffic is packetized and transferred in packets. The echo canceller removes echos, the DTMF detector detects the presence of keypad signals, the VAD detects the presence of voice, and the voice encoders compress the voice traffic into packets. However, when system resources are overtaxed and complexity reductions are required there are at least two methods for controlling the voice encoder. In the first method, the complexity level for the current frame is estimated from the information contained within previous voice frames and from the information gained from the echo canceller on the current voice frame. The resource manager then mandates complexity reductions for the processing of frames in the current frame interval in accordance with these estimations.

Alternatively, the voice encoders may be divided into a "front end" and a "back end". The front end performs voice activity detection and open loop pitch detection (in the case of G.723.1 and G.729 Annex A) on all channels operating on the DSP. Subsequent to the execution of the front end function for all channels of a particular voice encoder, the system complexity may be estimated based on the known information. Complexity reductions may then be mandated to ensure that the current processing cycle can satisfy the processing requirements of the voice encoders and decoders. This alternative method is preferred because the state of the VAD is known whereas in the previously described method the state of the VAD is estimated.

In the alternate method, once the front end processing is complete so that the state of the VAD and the voicing state for all channels is known, the system complexity may be estimated based on the known statistics for the current frame. In the first method, the state of the VAD and the voicing state may be estimated based on available known information. For example, the echo canceller processes a voice encoder input signal to remove line echos prior to the activation of the voice encoder. The echo canceller may estimate the state of the VAD based on the power level of a reference signal and the voice encoder input signal so that the complexity level of all controllable PXDs and services may be updated to determine the estimated complexity level of each assuming no complexity reductions have been invoked. If the sum of all the various complexity estimates is less than the complexity budget, no complexity reductions are required. Otherwise, the complexity level of all system components are estimated assuming the invocation of the transparent complexity reduction method to determine the estimated complexity resources required for the current processing frame. If the sum of the complexity estimates with transparent complexity reductions in place is less than the complexity budget, then the transparent complexity reduction is used for that frame. In a similar manner, more and more severe complexity reduction is considered until system complexity satisfies the prescribed budget.

37367/CAG/B600

The operating system should preferably allow processing to exceed the real-time constraint, i.e. maximum processing capability for the underlying DSP, in the short term. Thus data that should normally be processed within a given time frame or cycle may be buffered and processed in the next sequence. However, the overall complexity or processor loading must remain (on average) within the real-time constraint. This is a tradeoff between delay/jitter and channel density. Since packets may be delayed (due to processing overruns) overall end to end delay may increase slightly to account for the processing jitter.

Referring to FIG. 11, a preferred echo canceller has been modified to include an echo canceller bypass switch that invokes an echo suppressor in lieu of echo cancellation under certain system conditions so as to reduce processor loading. In addition, in the described exemplary embodiment the resource manager may instruct the adaptation logic 136 to disable filter adapter 134 so as to reduce processor loading under real-time constraints. The system will preferably limit adaptation on a fair and equitable basis when processing overruns occur. For example, if four echo cancellers are adapting when a processing over run occurs, the resource manager may disable the adaption of echo cancellers one and two. If the processing over run continues, the resource manger should preferably enable adaption of echo cancellers one and two, and reduce system complexity by disabling the adaptation of echo cancellers three and four. This limitation should preferably be adjusted such that channels which are fully adapted have adaptation disabled first. In the described exemplary embodiment, the operating systems should preferably control the subfunctions to limit peak system complexity. The subfunctions should be co-operative and include modifications to the echo canceller and the speech encoders.

B. <u>The Fax Relay Mode</u>

The transfer of fax signals over packet based networks may be accomplished by at least three alternative methods. In the first method, fax data signals are exchanged in real time. Typically, the sending and receiving fax machines are spoofed to allow transmission delays plus jitter of up to about 1.2 seconds. The second, store and forward mode, is a non real time method of transferring fax data signals. Typically, the fax communication is transacted locally, stored into memory and transmitted to the destination fax machine at a subsequent time. The third mode is a combination of store and forward mode with minimal spoofing to provide an approximate emulation of a typical fax connection.

In the fax relay mode, the network VHD invokes the packet fax data exchange. The packet fax data exchange provides demodulation and re-modulation of fax data signals. This approach results in considerable bandwidth savings since only the underlying unmodulated data signals are transmitted across the packet based network. The packet fax data exchange also

37367/CAG/B600

provides compensation for network jitter with a jitter buffer similar to that invoked in the packet voice exchange. Additionally, the packet fax data exchange compensates for lost data packets with error correction processing. Spoofing may also be provided during various stages of the procedure between the fax machines to keep the connection alive.

The packet fax data exchange is divided into two basic functional units, a demodulation system and a re-modulation system. In the demodulation system, the network VHD couples fax data signals from a circuit switched network, or a fax machine, to the packet based network. In the re-modulation system, the network VHD couples fax data signals from the packet network to the switched circuit network, or a fax machine directly.

During real time relay of fax data signals over a packet based network, the sending and receiving fax machines are spoofed to accommodate network delays plus jitter. Typically, the packet fax data exchange can accommodate a total delay of up to about 1.2 seconds. Preferably, the packet fax data exchange supports error correction mode (ECM) relay functionality, although a full ECM implementation is typically not required. In addition, the packet fax data exchange should preferably preserve the typical call duration required for a fax session over a PSTN/ISDN when exchanging fax data signals between two terminals.

The packet fax data exchange for the real time exchange of fax data signals between a circuit switched network and a packet based network is shown schematically in FIG. 24. In this exemplary embodiment, a connecting PXD (not shown) connecting the fax machine to the switch board 32' is transparent, although those skilled in the art will appreciate that various signal conditioning algorithms could be programmed into PXD such as echo cancellation and gain.

After the PXD (not shown), the incoming fax data signal 390a is coupled to the demodulation system of the packet fax data exchange operating in the network VHD via the switchboard 32'. The incoming fax data signal 390a is received and buffered in an ingress media queue 390. A V.21 data pump 392 demodulates incoming T.30 message so that T.30 relay logic 394 can decode the received T.30 messages 394a. Local T.30 indications 394b are packetized by a packetization engine 396 and if required, translated into T.38 packets via a T.38 shim 398 for transmission to a T.38 compliant remote network gateway (not shown) across the packet based network. The V.21 data pump 392 is selectively enabled/disabled 394c by the T.30 relay logic 394 in accordance with the reception/ transmission of the T.30 messages or fax data signals. The V.21 data pump 392 is common to the demodulation and re-modulation system. The V.21 data pump 392 communicates T.30 messages such as for example called station tone (CED) and calling station tone (CNG) to support fax setup between a local fax device (not shown) and a remote fax device (not shown) via the remote network gateway.

37367/CAG/B600

The demodulation system further includes a receive fax data pump 400 which demodulates the fax data signals during the data transfer phase. The receive fax data pump 400 supports the V.27ter standard for fax data signal transfer at 2400/4800 bps, the V.29 standard for fax data signal transfer at 7200/9600 bps, as well as the V.17 standard for fax data signal transfer at 7200/9600/12000/14400 bps. The V.34 fax standard, once approved, may also be supported. The T.30 relay logic 394 enables / disables 394d the receive fax data pump 400 in accordance with the reception of the fax data signals or the T.30 messages.

If error correction mode (ECM) is required, receive ECM relay logic 402 performs high level data link control(HDLC)de-framing, including bit de-stuffing and preamble removal on ECM frames contained in the data packets. The resulting fax data signals are then packetized by the packetization engine 396 and communicated across the packet based network. The T.30 relay logic 394 selectively enables / disables 394e the receive ECM relay logic 402 in accordance with the error correction mode of operation.

In the re-modulation system, if required, incoming data packets are first translated from a T.38 packet format to a protocol independent format by the T.38 packet shim 398. The data packets are then de-packetized by a depacketizing engine 406. The data packets may contain T.30 messages or fax data signals. The T.30 relay logic 394 reformats the remote T.30 indications 394f and forwards the resulting T.30 indications to the V.21 data pump 392. The modulated output of the V.21 data pump 392 is forwarded to an egress media queue 408 for transmission in either analog format or after suitable conversion, as 64 kbps PCM samples to the local fax device over a circuit switched network, such as for example a PSTN line.

De-packetized fax data signals are transferred from the depacketizing engine 406 to a jitter buffer 410. If error correction mode (ECM) is required, transmitting ECM relay logic 412 performs HDLC de-framing, including bit stuffing and preamble addition on ECM frames. The transmitting ECM relay logic 412 forwards the fax data signals, (in the appropriate format) to a transmit fax data pump 414 which modulates the fax data signals and outputs 8 KHz digital samples to the egress media queue 408. The T.30 relay logic selectively enables/disables (394g) the transmit ECM relay logic 412 in accordance with the error correction mode of operation.

The transmit fax data pump 414 supports the V.27ter standard for fax data signal transfer at 2400/4800 bps, the V.29 standard for fax data signal transfer at 7200/9600 bps, as well as the V.17 standard for fax data signal transfer at 7200/9600/12000/14400 bps. The T.30 relay logic selectively enables/disables (394h) the transmit fax data pump 414 in accordance with the transmission of the fax data signals or the T.30 message samples.

37367/CAG/B600

If the jitter buffer 410 underflows, a buffer low indication 410a is coupled to spoofing logic 416. Upon receipt of a buffer low indication during the fax data signal transmission, the spoofing logic 416 inserts "spoofed data" at the appropriate place in the fax data signals via the transmit fax data pump 414 until the jitter buffer 410 is filled to a pre-determined level, at which time the fax data signals are transferred out of the jitter buffer 410. Similarly, during the transmission of the T.30 message indications, the spoofing logic 416 can insert "spoofed data" at the appropriate place in the T.30 message samples via the V.21 data pump 392.

1. Data Rate Management

An exemplary embodiment of the packet fax data exchange complies with the T.38 recommendations for real-time Group 3 facsimile communication over packet based networks. In accordance with the T.38 standard, the preferred system should therefore, provide packet fax data exchange support at both the T.30 level (see ITU Recommendation T.30 - "Procedures for Document Facsimile Transmission in the General Switched Telephone Network", 1988) and the T4 level (see ITU Recommendation T.4 - "Standardization of Group 3 Facsimile Apparatus For Document Transmission", 1998), the contents of each of these ITU recommendations being incorporated herein by reference as if set forth in full. One function of the packet fax data exchange is to relay the set up (capabilities) parameters in a timely fashion. Spoofing may be needed at either or both the T.30 and T.4 levels to maintain the fax session while set up parameters are negotiated at each of the network gateways and relayed in the presence of network delays and jitter.

In accordance with the industry T.38 recommendations for real time Group 3 communication over packet based networks, the described exemplary embodiment relays all information including; T.30 preamble indications (flags), T.30 message data, as well as T.30 image data between the network gateways. The T.30 relay logic 394 in the sending and receiving network gateways then negotiate parameters as if connected via a PSTN line. The T.30 relay logic 394 interfaces with the V.21 data pump 392 and the receive and transmit data pumps 400 and 414 as well as the packetization engine 396 and the depacketizing engine 406 to ensure that the sending and the receiving fax machines 380(a) and 380(b) successfully and reliably communicate. The T.30 relay logic 394 provides local spoofing, using command repeats (CRP), and automatic repeat request (ARQ) mechanisms, incorporated into the T.30 protocol, to handle delays associated with the packet based network. In addition, the T.30 relay logic 394 intercepts control messages to ensure compatibility of the rate negotiation between the near end and far end machines including HDLC processing, as well as lost packet recovery according to the T.30 ECM standard.

FIG. 25 demonstrates message flow over a packet based network between a sending fax machine 380a (see FIG. 25) and the receiving fax device 380b (see FIG. 25) in non-ECM mode. The PSTN fax call is divided into five phases: call establishment, control and capabilities exchange, page transfer, end of page and multi-page signaling and call release. In the call establishment phase, the sending fax machine dials the sending network gateway 378a (see FIG. 25) which forwards calling tone (CNG) (not shown) to the receiving network gateway 378b (see FIG. 25). The receiving network gateway responds by alerting the receiving fax machine. The receiving fax machine answers the call and sends called station (CED) tones. The CED tones are detected by the V.21 data pump 392 of the receiving network gateway which issues an event 420 indicating the receipt of CED which is then relayed to the sending network gateway. The sending network gateway forwards the CED tone 422 to the sending fax device. In addition, the V.21 data pump of the receiving network gateway invokes the packet fax data exchange.

In the control and capabilities exchange, the receiving network gateway transmits T.30 preamble (HDLC flags) 424 followed by called subscriber identification (CSI) 426 and digital identification signal (DIS) 428 message which contains the capabilities of the receiving fax device. The sending network gateway, forwards the HDLC flags, CSI and DIS to the sending fax device. Upon receipt of CSI and DIS, the sending fax device determines the conditions for the call by examining its own capabilities table relative to those of the receiving fax device. The sending fax device issues a command to the sending network gateway 430 to begin transmitting HDLC flags. Next, the sending fax device transmits subscriber identification (TSI) 432 and digital command signal (DCS) 434 messages, which define the conditions of the call to the sending network gateway. In response, the sending network gateway forwards V.21 HDLC sending subscriber identification / frame check sequences and digital command signal / frame check sequences to the receiving fax device via the receiving network gateway. Next the sending fax device transmits training check (TCF) fields 436 to verify the training and ensure that the channel is suitable for transmission at the accepted data rate.

The TCF 436 may be managed by one of two methods. The first method, referred to as the data rate management method one in the T.38 standard, the receiving network gateway locally generate TCF. Confirmation to receive (CFR) is returned to the sending fax device 380(a), when the sending network gateway receives a confirmation to receive (CFR) 438 from the receiving fax machine via the receiving network gateway, and the TCF training 436 from the sending fax machine is received successfully. In the event that the receiving fax machine receives a CFR and the TCF training 436 from the sending fax machine subsequently fails, then DCS 434 from the sending fax machine is again relayed to the receiving fax machine. The TCF training 436 is repeated until an appropriate rate is established which provides successful TCF training 436 at both ends of the network.

37367/CAG/B600

In a second method to synchronize the data rate, referred to as the data rate management method two in the T.38 standard, the TCF data sequence received by the sending network gateway is forwarded from the sending fax machine to the receiving fax machine via the receiving network gateway. The sending and receiving fax machines then perform speed selection as if connected via a regular PSTN.

Upon receipt of confirmation to receive (CFR) 440 which indicates that all capabilities and the modulation speed have been confirmed, the sending fax machine enters the page transfer phase, and transmits image data 444 along with its training preamble 442. The sending network gateway receives the image data and forwards the image data 444 to the receiving network gateway. The receiving network gateway then sends its own training preamble 446 followed by the image data 448 to the receiving fax machine.

In the end of page and multi-page signaling phase, after the page has been successfully transmitted, the sending fax device sends an end of procedures (EOP) 450 message if the fax call is complete and all pages have been transmitted. If only one of multiple pages has been successfully transmitted, the sending fax device transmits a multi-page signal (MPS). The receiving fax device responds with message confirmation (MCF) 452 to indicate the message has been successfully received and that the receiving fax device is ready to receive additional pages. The release phase is the final phase of the call, where at the end of the final page, the receiving fax machine sends a message confirmation (MCF) 452, which prompts the sending fax machine to transmit a disconnect (DCN) signal 454. The call is then terminated at both ends of the network.

ECM fax relay message flow is similar to that described above. All preambles, messages and page transfers (phase C) HDLC data are relayed through the packet based network. Phase C HDLC data is de-stuffed and, along with the preamble and frame checking sequences (FCS), removed before being relayed so that only fax image data itself is relayed over the packet based network. The receiving network gateway performs bit stuffing and reinserts the preamble and FCS.

2. Spoofing Techniques

Spoofing refers to the process by which a facsimile transmission is maintained in the presence of data packet under-run due to severe network jitter or delay. An exemplary embodiment of the packet fax data exchange complies with the T.38 recommendations for real-time Group 3 facsimile communication over packet based networks. In accordance with the T.38 recommendations, a local and remote T.30 fax device communicate across a packet based

network via signal processing systems, which for the purposes of explanation are operating in network gateways. In operation, each fax device establishes a facsimile connection with its respective network gateway in accordance with the ITU-T.30 standards and the signal processing systems operating in the network gateways relay data signals across a packet based network.

In accordance with the T.30 protocol, there are ceratin time constraints on the handshaking and image data transmission for the facsimile connection between the T.30 fax device and its respective network gateway. The problem that arises is that the T.30 facsimile protocol is not designed to accommodate the significant jitter and packet delay that is common to communications across packet based networks. To prevent termination of the fax connection due to severe network jitter or delay, it is, therefore, desirable to ensure that both T.30 fax devices can be spoofed during periods of data packet under-run. FIG. 26 demonstrates fax communication 466 under the T.30 protocol, wherein a handshake negotiator 468, typically a low speed modem such as V.21, performs handshake negotiation and fax image data is communicated via a high speed data pump 470 such as V.27, V.29 or V.17. In addition, fax image data can be transmitted in an error correction mode (ECM) 472 or non error correction mode (non-ECM) 474, each of which uses a different data format.

Therefore, in the described exemplary embodiment, the particular spoofing technique utilized is a function of the transmission format. In the described exemplary embodiment, HDLC preamble 476 is used to spoof the T.30 fax devices during V.21 handshaking and during transmission of fax image data in the error correction mode. However, zero-bit filling 478 is used to spoof the T.30 fax devices during fax image data transfer in the non error correction mode. Although fax relay spoofing is described in the context of a signal processing system with the packet data fax exchange invoked, those skilled in the art will appreciate that the described exemplary fax relay spoofing method is likewise suitable for various other telephony and telecommunications application. Accordingly, the described exemplary embodiment of fax relay spoofing in a signal processing system is by way of example only and not by way of limitation.

a. <u>V.21 HDLC Preamble Spoofing</u>

The T.30 relay logic 394 packages each message or command into a HDLC frame which includes preamble flags. An HDLC frame structure is utilized for all binary-coded V.21 facsimile control procedures. The basic HDLC structure consists of a number of frames, each of which is subdivided into a number of fields. The HDLC frame structure provides for frame labeling and error checking. When a new facsimile transmission is initiated, HDLC preamble

in the form of synchronization sequences are transmitted prior to the binary coded information. The HDLC preamble is V.21 modulated bit streams of "01111110 (0x7e)".

In the described exemplary embodiment, spoofing techniques are utilized at the T.30 and T.4 levels to manage extended network delays and jitter. Turning back to FIG. 24, the T.30 relay logic 394 waits for a response to any message or command transmitted across the packet based network before continuing to the next state or phase. In accordance with an exemplary spoofing technique, the sending and receiving network gateways 378a, 378b (See FIG. 25) spoof their respective fax machines 380a, 380b by locally transmitting HDLC preamble flags if a response to a transmitted message is not received from the packet based network within approximately 1.5-2.0 seconds. The maximum length of the preamble is limited to about four seconds. If a response from the packet based network arrives before the spoofing time out, each network gateway should preferably transmit a response message to its respective fax machine following the preamble flags. Otherwise, if the network response to a transmitted message is not received prior to the spoofing time out (in the range of about 5.5-6.0 seconds), the response is assumed to be lost. In this case, when the network gateway times out and terminates preamble spoofing, the local fax device transmits the message command again. Each network gateway repeats the spoofing technique until a successful handshake is completed or its respective fax machine disconnects.

b. <u>ECM HDLC Preamble Spoofing</u>

The packet fax data exchange utilizes an HDLC frame structure for ECM high-speed data transmission. Preferably, the frame image data is divided by one or more HDLC preamble flags. If the network under-runs due to jitter or packet delay, the network gateways spoof their respective fax devices at the T.4 level by adding extra HDLC flags between frames. This spoofing technique increases the sending time to compensate for packet under-run due to network jitter and delay. Returning to FIG. 24 if the jitter buffer 410 underflows, a buffer low indication 410a is coupled to the spoofing logic 416. Upon receipt of a buffer low indication during the fax data signal transmission, the spoofing logic 416 inserts HDLC preamble flags at the frame boundary via the transmit fax data pump 414. When the jitter buffer 410 is filled to a pre-determined level, the fax image data is transferred out of the jitter buffer 410.

In the described exemplary embodiment, the jitter buffer 410 must be sized to store at least one HDLC frame so that a frame boundary may be located. The length of the largest T.4 ECM HDLC frame is 260 octets or 130 16-bit words. Spoofing is preferably activated when the number of packets stored in the jitter buffer 410 drops to a predetermined threshold level. When spoofing is required, the spoofing logic 416 adds HDLC flags at the frame boundary as a

37367/CAG/B600

complete frame is being reassembled and forwarded to the transmit fax data pump 414. This continues until the number of data packets in the jitter buffer 410 exceeds the threshold level. The maximum time the network gateways will spoof their respective local fax devices can vary but can generally be about ten seconds.

c. Non-ECM Spoofing with Zero Bit Filling

T.4 spoofing handles delay impairments during page transfer or C phase of a fax call. For those systems that do not utilize ECM, phase C signals comprise a series of coded image data followed by fill bits and end-of-line (EOL) sequences. Typically, fill bits are zeros inserted between the fax data signals and the EOL sequences, "000000000001". Fill bits ensure that a fax machine has time to perform the various mechanical overhead functions associated with any line it receives. Fill bits can also be utilized to spoof the jitter buffer to ensure compliance with the minimum transmission time of the total coded scan line established in the pre-message V.21 control procedure. The number of the bits of coded image contained in the data signals associated with the scan line and transmission speed limit the number of fill bits that can be added to the data signals. Preferably, the maximum transmission of any coded scan line is limited to less than about 5 sec. Thus, if the coded image for a given scan line contains 1000 bits and the transmission rate is 2400 bps, then the maximum duration of fill time is (5 -(1000 +12)/2400) = 4.57 sec.

Generally, the packet fax data exchange utilizes spoofing if the network jitter delay exceeds the delay capability of the jitter buffer 410. In accordance with the EOL spoofing method, fill bits can only be inserted immediately before an EOL sequence, so that the jitter buffer 410 should preferably store at least one EOL sequence. Thus the jitter buffer 410 should preferably be sized to hold at least one entire scan line of data to ensure the presence of at least one EOL sequence within the jitter buffer 410. Thus, depending upon transmission rate, the size of the jitter buffer 410 can become prohibitively large. The table below summarizes the desired jitter buffer data space to perform EOL spoofing for various scan line lengths. The table assumes that each pixel is represented by a single bit. The values represent an approximate upper limit on the required data space, but not the absolute upper limit, because in theory at least, the longest scan line can consist of alternating black and white pixels which would require an average of 4.5 bits to represent each pixel rather than the one to one ratio summarized in the table.

Scan Line Length	Number of words	sec to print out at 2400	sec to print out at 4800	sec to print out at 9600	sec to print out at 14400
1728	108	0.72	0.36	0.18	0.12

37367/CAG/B600

2048	128	0.853	0.427	0.213	0.14
2432	152	1.01	0.507	0.253	0.17
3456	216	1.44	0.72	0.36	0.24
4096	256	2	0.853	0.43	0.28
4864	304	2.375	1.013	0.51	0.34

To ensure the jitter buffer 410 stores an EOL sequence, the spoofing logic 416 should be activated when the number of data packets stored in the jitter buffer 410 drops to a threshold level. Typically, a threshold value of about 200 msec is used to support the most commonly used fax setting, namely a fax speed of 9600 bps and scan line length of 1728. An alternate spoofing method should be used if an EOL sequence is not contained within the jitter buffer 410, otherwise the call will have to be terminated. An alternate spoofing method uses zero run length code words. This method requires real time image data decoding so that the word boundary is known. Advantageously, this alternate method reduces the required size of the jitter buffer 410.

Simply increasing the storage capacity of the jitter buffer 410 can minimize the need for spoofing. However, overall network delay increases when the size of the jitter buffer 410 is increased. Increased network delay may complicate the T.30 negotiation at the end of page or end of document, because of susceptibility to time out. Such a situation arises when the sending fax machine completes the transmission of high speed data, and switches to an HDLC phase and sends the first V.21 packet in the end of page / multi-page signaling phase, (i.e. phase D). The sending fax machine must be kept alive until the response to the V.21 data packet is received. The receiving fax device requires more time to flush a large jitter buffer and then respond, hence complicating the T.30 negotiation.

In addition, the length of time a fax machine can be spoofed is limited, so that the jitter buffer 410 can not be arbitrarily large. A pipeline store and forward relay is a combination of store and forward and spoofing techniques to approximate the performance of a typical Group 3 fax connection when the network delay is large (on the order of seconds or more). One approach is to store and forward a single page at a time. However, this approach requires a significant amount of memory (10 Kwords or more). One approach to reduce the amount of memory required entails discarding scan lines on the sending network gateway and performing line repetition on the receiving network gateway so as to maintain image aspect ratio and quality. Alternatively, a partial page can be stored and forwarded thereby reducing the required amount of memory.

37367/CAG/B600

The sending and receiving fax machines will have some minimal differences in clock frequency. ITU standards recommends a data pump data rate of ± 100 ppm, so that the clock frequencies between the receiving and sending fax machines could differ by up to 200 ppm. Therefore, the data rate at the receiving network gateway (jitter buffer 410) can build up or deplete at a rate of 1 word for every 5000 words received. Typically a fax page is less than 1000 words so that end to end clock synchronization is not required.

C. Data Relay Mode

In the data relay mode, the packet data modem exchange provides demodulation and modulation of data signals. With full duplex capability, both modulation and demodulation of data signals can be performed simultaneously. The packet data modem exchange also provides compensation for network jitter with a jitter buffer similar to that invoked in the packet voice exchange. Additionally, the packet data modem exchange compensates for system clock jitter between modems with a dynamic phase adjustment and resampling mechanism. Spoofing may also be provided during various stages of the call negotiation procedure between the modems to keep the connection alive.

The packet data modem exchange invoked by the network VHD in the data relay mode is shown schematically in FIG. 27. In the described exemplary embodiment, a connecting PXD (not shown) connecting a modem to the switch board 32' is transparent, although those skilled in the art will appreciate that various signal conditioning algorithms could be programmed into the PXD such as filtering, echo cancellation and gain.

After the PXD, the data signals are coupled to the network VHD via the switchboard 32'. The packet data modem exchange provides two way communication between a circuit switched network and packet based network with two basic functional units, a demodulation system and a remodulation system. In the demodulation system, the network VHD exchanges data signals from a circuit switched network, or a telephony device directly, to a packet based network. In the remodulation system, the network VHD exchanges data signals from the packet based network to the PSTN line, or the telephony device.

In the demodulation system, the data signals are received and buffered in an ingress media queue 500. A data pump receiver 504 demodulates the data signals from the ingress media queue 500. The data pump receiver 504 supports the V.22bis standard for the demodulation of data signals at 1200/2400 bps; the V.32bis standard for the demodulation of data signals at 4800/7200/9600/12000/14400 bps, as well as the V.34 standard for the demodulation of data signals up to 33600 bps. Moreover, the V.90 standard may also be supported. The demodulated

37367/CAG/B600

data signals are then packetized by the packetization engine 506 and transmitted across the packet based network.

In the remodulation system, packets of data signals from the packet based network are first depacketized by a depacketizing engine 508 and stored in a jitter buffer 510. A data pump transmitter 512 modulates the buffered data signals with a voiceband carrier. The modulated data signals are in turn stored in the egress media queue 514 before being output to the PXD (not shown) via the switchboard 32'. The data pump transmitter 512 supports the V.22bis standard for the transfer of data signals at 1200/2400 bps; the V.32bis standard for the transfer of data signals at 4800/7200/9600/12000/14400 bps, as well as the V.34 standard for the transfer of data signal up to 33600 bps. Moreover, the V.90 standard may also be supported.

During jitter buffer underflow, the jitter buffer 510 sends a buffer low indication 510a to spoofing logic 516. When the spoofing logic 516 receives the buffer low signal indicating that the jitter buffer 510 is operating below a predetermined threshold level, it inserts spoofed data at the appropriate place in the data signal via the data pump transmitter 512. Spoofing continues until the jitter buffer 510 is filled to the predetermined threshold level, at which time data signals are again transferred from the jitter buffer 510 to the data pump transmitter 512.

End to end clock logic 518 also monitors the state of the jitter buffer 510. The clock logic 518 controls the data transmission rate of the data pump transmitter 512 in correspondence to the state of the jitter buffer 510. When the jitter buffer 510 is below a predetermined threshold level, the clock logic 518 reduces the transmission rate of the data pump transmitter 512. Likewise, when the jitter buffer 510 is above a predetermined threshold level, the clock logic 518 increases the transmission rate of the data pump transmitter 512.

Before the transmission of data signals across the packet based network, the connection between the two modems must first be negotiated through a handshaking sequence. This entails a two-step process. First, a call negotiator 502 determines the type of modem (i.e., V.22, V.32bis, V.34, V.90, etc.) connected to each end of the packet based network. Second, a rate negotiator 520 negotiates the data signal transmission rate between the two modems.

The call negotiator 502 determines the type of modem connected locally, as well as the type of modem connected remotely via the packet based network. The call negotiator 502 utilizes V.25 automatic answering procedures and V.8 auto-baud software to automatically detect modem capability. The call negotiator 502 receives protocol indication signals 502a (ANSam and V.8 menus) from the ingress media queue 500, as well as AA, AC and other message indications 502b from the local modem via a data pump state machine 522, to determine the type

of modem in use locally. The call negotiator 502 relays the ANSam answer tones and other indications 502e from the data pump state machine 522 to the remote modem via a packetization engine 506. The call negotiator also receives ANSam, AA, AC and other indications 502c from a remote modem (not shown) located on the opposite end of the packet based network via a depacketizing engine 508. The call negotiator 502 relays ANSam answer tones and other indications 502d to a local modem (not shown) via an egress media queue 514 of the modulation system. With the ANSam, AA, AC and other indications from the local and remote modems, the call negotiator 502 can then negotiate a common standard (i.e., V.22, V.32bis, V.34, V.90, etc.) in which the data pumps must communicate with the local modem and the remote modems.

The packet data modem exchange preferably utilizes indication packets as a means for communicating answer tones, AA, AC and other indication signals across the packet based network However, the packet data modem exchange supports data pumps such as V.22bis and V.32bis which do not include a well defined error recovery mechanism, so that the modem connection may be terminated whenever indication packets are lost. Therefore, either the packet data modem exchange or the application layer should ensure proper delivery of indication packets when operating in a network environment that does not guarantee packet delivery.

The packet data modem exchange can ensure delivery of the indication packets by periodically retransmitting the indication packet until some expected packets are received. For example, in V.32bis relay, the call negotiator operating under the packet data modem exchange on the answer network gateway periodically retransmits ANSam answer tones from the answer modem to the call modem, until the calling modem connects to the line and transmits carrier state AA.

Alternatively, the packetization engine can embed the indication information directly into the packet header. In this approach, an alternate packet format is utilized to include the indication information. During modem handshaking, indication packets transmitted across the packet based network include the indication information, so that the system does not rely on the successful transmission of individual indication packets. Rather, if a given packet is lost, the next arriving packet contains the indication information in the packet header. Both methods increase the traffic across the network. However, it is preferable to periodically retransmit the indication packets because it has less of a detrimental impact on network traffic.

A rate negotiator 520 synchronizes the connection rates at the network gateways 496a, 496b, 496c (see FIG. 29). The rate negotiator receives rate control codes 520a from the local modem via the data pump state machine 522 and rate control codes 520b from the remote modem via the depacketizing engine 508. The rate negotiator 520 also forwards the remote rate control

37367/CAG/B600

codes 520a received from the remote modem to the local modem via commands sent to the data pump state machine 522. The rate negotiator 520 forwards the local rate control codes 520c received from the local modem to the remote modem via the packetization engine 506. Based on the exchanged rate codes the rate negotiator 520 establishes a common data rate between the calling and answering modems. During the data rate exchange procedure, the jitter buffer 510 should be disabled by the rate negotiator 520 to prevent data transmission between the call and answer modems until the data rates are successfully negotiated.

Similarly error control (V.42) and data compression (V.42bis) modes should be synchronized at each end of the packet based network. Error control logic 524 receives local error control messages 524a from the data pump receiver 504 and forwards those V.14/V.42 negotiation messages 524c to the remote modem via the packetization engine 506. In addition, error control logic 524 receives remote V.14/V.42 indications 524b from the depacketizing engine 508 and forwards those V.14/V.42 indications 524d to the local modem. With the V.14/V.42 indications from the local and remote modems, the error control logic 524 can negotiate a common standard to ensure that the network gateways utilize a common error protocol. In addition, error control logic 524, communicates the negotiated error control protocol 524(e) to the spoofing logic 516 to ensure data mode spoofing is in accordance with the negotiated error control mode.

V.42 is a standard error correction technique using advanced cyclical redundancy checks and the principle of automatic repeat requests (ARQ). In accordance with the V.42 standard, transmitted data signals are grouped into blocks and cyclical redundancy calculations add error checking words to the transmitted data signal stream. The receiving modem calculates new error check information for the data signal block and compares the calculated information to the received error check information. If the codes match, the received data signals are valid and another transfer takes place. If the codes do not match, a transmission error has occurred and the receiving modem requests a repeat of the last data block. This repeat cycle continues until the entire data block has been received without error.

Various voiceband data modem standards exist for error correction and data compression. V.42bis and MNP5 are examples of data compression standards. The handshaking sequence for every modem standard is different so that the packet data modem exchange should support numerous data transmission standards as well as numerous error correction and data compression techniques.

37367/CAG/B600

1. End to End Clock Logic

Slight differences in the clock frequency of the call modem and the answer modem are expected, since the baud rate tolerance for a typical modem data pump is ±100 ppm . This tolerance corresponds to a relatively low depletion or build up rate of 1 in 5000 words. However, the length of a modem session can be very long, so that uncorrected difference in clock frequency may result in jitter buffer underflow or overflow.

In the described exemplary embodiment, the clock logic synchronizes the transmit clock of the data pump transmitter 512 to the average rate at which data packets arrive at the jitter buffer 510. The data pump transmitter 512 packages the data signals from the jitter buffer 510 in frames of data signals for demodulation and transmission to the egress media queue 514. At the beginning of each frame of data signals, the data pump transmitter 512 examines the egress media queue 514 to determine the remaining buffer space, and in accordance therewith, the data pump transmitter 512 modulates that number of digital data samples required to produce a total of slightly more or slightly less than 80 samples per frame, assuming that the data pump transmitter 512 is invoked once every 10 msec. The data pump transmitter 512 gradually adjusts the number of samples per frame to allow the receiving modem to adjust to the timing change. Typically, the data pump transmitter 512 uses an adjustment rate of about one ppm per frame. The maximum adjustment should be less than about 200 ppm.

In the described exemplary embodiment, end to end clock logic 518 monitors the space available within the jitter buffer 510 and utilizes water marks to determine whether the data rate of the data pump transmitter 512 should be adjusted. Network jitter may cause timing adjustments to be made. However, this should not adversely affect the data pump receiver of the answering modem as these timing adjustments are made very gradually.

2. Modem Connection Handshaking Sequence.

a. Call Negotiation.

A single industry standard for the transmission of modem data over a packet based network does not exist. However, numerous common standards exist for transmission of modem data at various data rates over the PSTN. For example, V.22 is a common standard used to define operation of 1200 bps modems. Data rates as high as 2400 bps can be implemented with the V.22bis standard (the suffix "bis" indicates that the standard is an adaptation of an existing standard). The V.22bis standard groups data signals into four bit words which are transmitted at 600 baud. The V.32 standard supports full duplex, data rates of up to 9600 bps over the PSTN.

37367/CAG/B600

A V.32 modem groups data signals into four bit words and transmits at 2400 baud. The V.32bis standard supports duplex modems operating at data rates up to 14,400 bps on the PSTN. In addition, the V.34 standard supports data rates up to 33,600 bps on the public switched telephone network. In the described exemplary embodiment, these standards can be used for data signal transmission over the packet based network with a call negotiator that supports each standard.

b. Rate Negotiation.

Rate negotiation refers to the process by which two telephony devices are connected at the same data rate prior to data transmission. In the context of a modem connection in accordance with an exemplary embodiment of the present invention, each modem is coupled to a signal processing system, which for the purposes of explanation is operating in a network gateway, either directly or through a PSTN line. In operation, each modem establishes a modem connection with its respective network gateway, at which point, the modems begin relaying data signals across a packet based network. The problem that arises is that each modem may negotiate a different data rate with its respective network gateway, depending on the line conditions and user settings. In this instance, the data signals transmitted from one of the modems will enter the packet based network faster than it can be extracted at the other end by the other modem. The resulting overflow of data signals may result in a lost connection between the two modems. To prevent data signal overflow, it is, therefore, desirable to ensure that both modems negotiate to the same data rate. A rate negotiator can be used for this purpose. Although the the rate negotiator is described in the context of a signal processing system with the packet data modem exchange invoked, those skilled in the art will appreciate that the rate negotiator is likewise suitable for various other telephony and telecommunications application. Accordingly, the described exemplary embodiment of the rate negotiator in a signal processing system is by way of example only and not by way of limitation.

In an exemplary embodiment, data rate negotiation is achieved through a data rate negotiation procedure, wherein a call modem independently negotiates a data rate with a call network gateway, and an answer modem independently negotiates a data rate with an answer network gateway. The calling and answer network gateways, each having a signal processing system running a packet exchange, then exchange data packets containing information on the independently negotiated data rates. If the independently negotiated data rates are the same, then each rate negotiator will enable its respective network gateway and data transmission between the call and answer modems will commence. Conversely, if the independently negotiated data rates are different, the rate negotiator will renegotiate the data rate by adopting the lowest of the two data rates. The call and answer modems will then undergo retraining or rate renegotiation

procedures by their respective network gateways to establish a new connection at the renegotiated data rate. The advantage of this approach is that the data rate negotiation procedure takes advantage of existing modem functionality, namely, the retraining and rate renegotiation mechanism, and puts it to alternative usage. Moreover, by retraining both the call and answer modem (one modem will already be set to the renegotiated rate) both modems are automatically prevented from sending data.

Alternatively, the calling and answer modems can directly negotiate the data rate. This method is not preferred for modems with time constrained handshaking sequences such as, for example, modems operating in accordance with the V.22bis or the V.32bis standards. The round trip delay accommodated by these standards could cause the modem connection to be lost due to timeout. Instead, retrain or rate renegotiation should be used for data signals transferred in accordance with the V.22bis and V.32bis standards, whereas direct negotiation of the data rate by the local and remote modems can be used for data exchange in accordance with the V.34 and V.90 (a digital modem and analog modem pair for use on PSTN lines at data rates up to 56,000 bps downstream and 33,600 upstream) standards.

c. <u>Exemplary Handshaking Sequences</u>.

(V.22 Handshaking Sequence)

The call negotiator on the answer network gateway, differentiates between modem types and relays the ANSam answer tone. The answer modem transmits unscrambled binary ones signal (USB1) indications to the answer mode gateway. The answer network gateway forwards USB1 signal indications to the call network gateway. The call negotiator in the call network gateway assumes operation in accordance with the V.22bis standard as a result of the USB1 signal indication and terminates the call negotiator. The packet data modem exchange, in the answer network gateway then invokes operation in accordance with the V.22bis standard after an answer tone timeout period and terminates its call negotiator.

V.22bis handshaking does not utilize rate messages or signaling to indicate the selected bit rate as with most high data rate pumps. Rather, the inclusion of a fixed duration signal (S1) indicates that 2400 bps operation is to be used. The absence of the S1 signal indicates that 1200 bps should be selected. The duration of the S1 signal is typically about 100 msec, making it likely that the call modem will perform rate determination (assuming that it selects 2400 bps) before rate indication from the answer modem arrives. Therefore, the rate negotiator in the call network gateway should select 2400 bps operation and proceed with the handshaking procedure. If the answer modem is limited to a 1200 bps connection, rate renegotiation is typically used to

37367/CAG/B600

change the operational data rate of the call modem to1200 bps. Alternatively, if the call modem selects 1200 bps, rate renegotiation would not be required.

(V.32bis Handshaking Sequence)

V32bis handshaking utilizes rate signals (messages) to specify the bit rate. A relay sequence in accordance with the V.32bis standard is shown in FIG. 28 and begins with the call negotiator in the answer network gateway relaying ANSam 530 answer tone from the answer modem to the call modem. After receiving the answer tone for a period of at least one second, the call modem connects to the line and repetitively transmits carrier state A 532. When the call network gateway detects the repeated transmission of carrier state A ("AA"), the call network gateway relays this information 534 to the answer network gateway. In response the answer network gateway forwards the AA indication to the answer modem and invokes operation in accordance with the V.32bis standard. The answer modem then transmits alternating carrier states A and C 536 to the answer network gateway. If the answer network gateway receives AC from the answer modem, the answer network gateway relays AC 538 to the call network gateway, thereby establishing operation in accordance with the V.32bis standard, allowing call negotiator in the call network gateway to be terminated. Next, data rate alignment is achieved by either of two methods.

In the first method for data rate alignment of a V.32bis relay connection, the call modem and the answer modem independently negotiate a data rate with their respective network gateways at each end of the network 540 and 542. Next, each network gateway forwards a connection data rate indication 544 and 546 to the other network gateway. Each network gateway compares the far end data rate to its own data rate. The preferred rate is the minimum of the two rates. Rate renegotiation 548 and 550 is invoked if the connection rate of either network gateway to its respective modem differs from the preferred rate.

In the second method, rate signals R1, R2 and R3, are relayed to achieve data rate negotiation. FIG. 29 shows a relay sequence in accordance with the V.32bis standard for this alternate method of rate negotiation. The call negotiator relays the answer tone (ANSam) 552 from the answer modem to the call modem. When the call modem detects answer tone, it repetitively transmits carrier state A 554 to the call network gateway. The call network gateway relays this information (AA) 556 to the answer network gateway. The answer network gateway sends the AA 558 to the answer modem and initiates normal range tone exchange with the answer modem. The answer network gateway then forwards AC 560 to call network gateway which in turn relays this information 562 to the call modem to initiate normal range tone exchange between the call network gateway and the call modem.

37367/CAG/B600

The answer modem sends its first training sequence 564 followed by R1 (the data rates currently available in the answer modem) to the rate negotiator in the answer network gateway. When the answer network gateway receives an R1 indication, it forwards R1 566 to the call network gateway. The answer network gateway then repetitively sends training sequences to the answer modem. The call network gateway forwards the R1 indication 570 of the answer modem to the call modem. The call modem sends training sequences to the call network gateway 572. The call network gateway determines the data rate capability of the call modem, and forwards the data rate capabilities of the call modem to the answer network gateway in a data rate signal format. The call modem also sends an R2 indication 568 (data rate capability of the call modem, preferably excluding rates not included in the previously received R1 signal, i.e. not supported by the answer modem) to the call network gateway which forwards it to the answer network gateway. The call network gateway then repetitively sends training sequences to the call modem until receiving an R3 signal 574 from the answer modem via the answer network gateway.

The answer network gateway performs a logical AND operation on the R1 signal from the answer modem (data rate capability of the answer modem), the R2 signal from the call modem (data rate capability of the call modem, excluding rates not supported by the answer modem) and the training sequences of the call network gateway (data rate capability of the call modem) to create a second rate signal R2 576, which is forwarded to the answer modem. The answer modem sends its second training sequence followed an R3 signal, which indicates the data rate to be used by both modems. The answer network gateway relays R3 574 to the call network gateway which forwards it to the call modem and begins operating at the R3 specified bit rate. However, this method of rate synchronization is not preferred for V.32bis due to time constrained handshaking.

(V.34 Handshaking Sequence)

Data transmission in accordance with the V.34 standard utilizes a modulation parameter (MP) sequence to exchange information pertaining to data rate capability. The MP sequences can be exchanged end to end to achieve data rate synchronization. Initially, the call negotiator in the answer network gateway relays the answer tone (ANSam) from the answer modem to the call modem. When the call modem receives answer tone, it generates a CM indication and forwards it to the call network gateway. When the call network gateway receives a CM indication, it forwards it to the answer network gateway which then communicates the CM indication with the answer modem. The answer modem then responds by transmitting a JM sequence to the answer network gateway, which is relayed by the answer network gateway to the call modem via the call network gateway. If the call network gateway then receives a CJ sequence from the call modem, the call negotiator in the call network gateway, initiates operation

in accordance with the V.34 standard, and forwards a CJ sequence to the answer network gateway. If the JM menu calls for V.34, the call negotiator in the answer network gateway initiates operation in accordance with the V.34 standard and the call negotiator is terminated. If a standard other than V.34 is called for, the appropriate procedure is invoked, such as those described previously for V.22 or V.32bis. Next, data rate alignment is achieved by either of two methods.

In a first method for data rate alignment after a V.34 relay connection is established, the call modem and the answer modem freely negotiate a data rate at each end of the network with their respective network gateways. Each network gateway forwards a connection rate indication to the other gateway. Each gateway compares the far end bit rate to the rate transmitted by each gateway. For example, the call network gateway compares the data rate indication received from the answer modem gateway to that which it negotiated freely negotiated to with the call modem. The preferred rate is the minimum of the two rates. Rate renegotiation is invoked if the connection rate at the calling or receiving end differs from the preferred rate, to force the connection to the desired rate.

In an alternate method for V.34 rate synchronization, MP sequences are utilized to achieve rate synchronization without rate renegotiation. The call modem and the answer modem independently negotiate with the call network gateway and the answer network gateway respectively until phase IV of the negotiations is reached . The call network gateway and the answer network gateway exchange training results in the form of MP sequences when Phase IV of the independent negotiations is reached to establish the primary and auxiliary data rates. The call network gateway and the answer network gateway are preferably prevented from relaying MP sequences to the call modem and the answer modem respectively until the training results for both network gateways and the MP sequences for both modems are available. If symmetric rate is enforced, the maximum answer data rate and the maximum call data rate of the four MP sequences are compared. The lower data rate of the two maximum rates is the preferred data rate. Each network gateway sends the MP sequence with the preferred rate to its respective modem so that the calling and answer modems operate at the preferred data rate.

If asymmetric rates are supported, then the preferred call-answer data rate is the lesser of the two highest call-answer rates of the four MP sequences. Similarly, the preferred answer-call data rate is the lesser of the two highest answer-call rates of the four MP sequences. Data rate capabilities may also need to be modified when the MP sequence are formed so as to be sent to the calling and answer modems. The MP sequence sent to the calling and answer modems, is the logical AND of the data rate capabilities from the four MP sequences.

37367/CAG/B600

(V.90 Handshaking Sequence)

The V.90 standard utilizes a digital and analog modem pair to transmit modem data over the PSTN line. The V.90 standard utilizes MP sequences to convey training results from a digital to an analog modem, and a similar sequence, using constellation parameters (CP) to convey training results from an analog to a digital modem. Under the V.90 standard, the timeout period is 15 seconds compared to a timeout period of 30 seconds under the V.34 standard. In addition, the analog modems control the handshake timing during training. In an exemplary embodiment, the call modem and the answer modem are the V.90 analog modems. As such the call modem and the answer modem are beyond the control of the network gateways during training. The digital modems only control the timing during transmission of TRN1d, which the digital modem in the network gateway uses to train its echo canceller.

When operating in accordance with the V.90 standard, the call negotiator utilizes the V.8 recommendations for initial negotiation. Thus, the initial negotiation of the V.90 relay session is substantially the same as the relay sequence described for V.34 rate synchronization method one and method two with asymmetric rate operation. There are two configurations where V.90 relay may be used. The first configuration is data relay between two V.90 analog modems, i.e. each of the network gateways are configured as V.90 digital modems. The upstream rate between two V.90 analog modems, according to the V.90 standard, is limited to 33,600 bps. Thus, the maximum data rate for an analog to analog relay is 33,600 bps. In accordance with the V.90 standard, the minimum data rate a V.90 digital modem will support is 28,800 bps. Therefore, the connection must be terminated if the maximum data rate for one or both of the upstream directions is less than 28,800 bps, and one or both the downstream direction is in V.90 digital mode. Therefore, the V.34 protocol is preferred over V.90 for data transmission between local and remote analog modems.

A second configuration is a connection between a V.90 analog modem and a V.90 digital modem. A typical example of such a configuration is when a user within a packet based PABX system dials out into a remote access server (RAS) or an Internet service provider (ISP) that uses a central site modem for physical access that is V.90 capable. The connection from PABX to the central site modem may be either through PSTN or directly through an ISDN, T1 or E1 interface. Thus the V.90 embodiment should preferably support an analog modem interfacing directly to ISDN, T1 or E1.

For an analog to digital modem connection, the connections at both ends of the packet based network should be either digital or analog to achieve proper rate synchronization. The analog modem decides whether to select digital mode as specified in INFO1a, so that INFO1a

should be relayed between the calling and answer modem via their respective network gateways before operation mode is synchronized.

Upon receipt of an INFO1a signal from the answer modem, the answer network gateway performs a line probe on the signal received from the answer modem to determine whether digital mode can be used. The call network gateway receives an INFO1a signal from the call modem. The call network gateway sends a mode indication to the answer network gateway indicating whether digital or analog will be used and initiates operation in the mode specified in INFO1a. Upon receipt of an analog mode indication signal from the call network gateway, the answer network gateway sends an INFO1a sequence to the answer modem. The answer network gateway then proceeds with analog mode operation. Similarly, if digital mode is indicated and digital mode can be supported by the answer modem, the answer network gateway sends an INFO1a sequence to the answer modem indicating that digital mode is desired and proceeds with digital mode operation.

Alternatively, if digital mode is indicated and digital mode can not be supported by the answer modem, the call modem should preferably be forced into analog mode by one of three alternate methods. First, some commercially available V.90 analog modems may revert to analog mode after several retrains. Thus, one method to force the call modem into analog mode is to force retrains until the call modem selects analog mode operation. In an alternate method, the call network gateway modifies its line probe so as to force the call modem to select analog mode. In a third method, the call modem and the answer modem operate in different modes. Under this method if the answer modem can not support a 28,800 bps data rate the connection is terminated.

3. Data Mode Spoofing

The jitter buffer 510 may underflow during long delays of data signal packets. Jitter buffer underflow can cause the data pump transmitter 512 to run out of data, and therefore, it is desirable that the jitter buffer 510 be spoofed with bit sequences. Preferably the bit sequences are benign. In the described exemplary embodiment, the specific spoofing methodology is dependent upon the common error mode protocol negotiated by the error control logic of each network gateway.

In accordance with V.14 recommendations, the spoofing logic 516 checks for character format and boundary (number of data bits, start bits and stop bits) within the jitter buffer 510. As specified in the V.14 recommendation the spoofing logic 516 must account for stop bits omitted due to asynchronous-to-synchronous conversion. Once the spoofing logic 516 locates the character boundary, ones can be added to spoof the local modem and keep the connection

37367/CAG/B600

alive. The length of time a modem can be spoofed with ones depends only upon the application program driving the local modem.

In accordance with the V.42 recommendations, the spoofing logic 516 checks for HDLC flag (HDLC frame boundary) within the jitter buffer 510. The basic HDLC structure consists of a number of frames, each of which is subdivided into a number of fields. The HDLC frame structure provides for frame labeling and error checking. When a new data transmission is initiated, HDLC preamble in the form of synchronization sequences are transmitted prior to the binary coded information. The HDLC preamble is modulated bit streams of "01111110 (0x7e)". The jitter buffer 510 should be sufficiently large to guarantee that at least one complete HDLC frame is contained within the jitter buffer 510. The default length of an HDLC frame is 132 octets. The V.42 recommendations for error correction of data circuit terminating equipment (DCE) using asynchronous-to-synchronous conversion does not specify a maximum length for an HDLC frame. However, because the length of the frame affects the overall memory required to implement the protocol, a information frame length larger than 260 octets is unlikely.

The spoofing logic 516 stores a threshold water mark (with a value set to be approximately equal to the maximum length of the HDLC frame). Spoofing is preferably activated when the number of packets stored in the jitter buffer 510 drops to the predetermined threshold level. When spoofing is required, the spoofing logic 516 adds HDLC flags at the frame boundary as a complete frame is being reassembled and forwarded to the transmit data pump. This continues until the number of data packets in the jitter buffer 510 exceeds the threshold level.

4. Retrain and Rate Renegotiation

In the described exemplary embodiment, if data rates independently negotiated between the modems and their respective network gateways are different, the rate negotiator will renegotiate the data rate by adopting the lowest of the two data rates. The call and answer modems will then undergo retraining or rate renegotiation procedures by their respective network gateways to establish a new connection at the renegotiated data rate. In addition, rate synchronization may be lost during a modem communication, requiring modem retraining and rate renegotiation, due to drift or change in the conditions of the communication channel. When a retrain occurs, an indication should be forwarded to the network gateway at the end of the packet based network. The network gateway receiving a retrain indication should initiate retrain with the connected modem to keep data flow in synchronism between the two connections. Rate synchronization procedures as previously described should be used to maintain data rate alignment after retrains.

37367/CAG/B600

Similarly, rate renegotiation causes both the calling and answer network gateways and to perform rate renegotiation. However, rate signals or MP (CP) sequences should be exchanged per method two of the data rate alignment as previously discussed for a V.32bis or V.34 rate synchronization whichever is appropriate.

5. Error Correcting Mode Synchronization

Error control (V.42) and data compression (V.42bis) modes should be synchronized at each end of the packet based network. In a first method, the call modem and the answer modem independently negotiate an error correction mode with each other on their own, transparent to the network gateways. This method is preferred for connections wherein the network delay plus jitter is relatively small, as characterized by an overall round trip delay of less than 700 msec.

Data compression mode is negotiated within V.42 so that the appropriate mode indication can be relayed when the calling and answer modems have entered into V.42 mode.

An alternative method is to allow modems at both ends to freely negotiate the error control mode with their respective network gateways. The network gateways must fully support all error correction modes when using this method. Also, this method cannot support the scenario where one modem selects V.14 while the other modem selects a mode other than V.14. For the case where V.14 is negotiated at both sides of the packet based network, an 8-bit no parity format is assumed by each respective network gateway and the raw demodulated data bits are transported there between. With all other cases, each gateway shall extract de-framed (error corrected) data bits and forward them to its counterpart at the opposite end of the network. Flow control procedures within the error control protocol may be used to handle network delay. The advantage of this method over the first method is its ability to handle large network delays and also the scenario where the local connection rates at the network gateways are different. However, packets transported over the network in accordance with this method must be guaranteed to be error free. This may be achieved by establishing a connection between the network gateways in accordance with the link access protocol connection for modems (LAPM)

6. Data Pump

Preferably, the data exchange includes a modem relay having a data pump for demodulating modem data signals from a modem for transmission on the packet based network, and remodulating modem data signal packets from the packet based network for transmission to a local modem. Similarly, the data exchange also preferably includes a fax relay with a data pump for demodulating fax data signals from a fax for transmission on the packet based network,

37367/CAG/B600

and remodulating fax data signal packets from the packet based network for transmission to a local fax device. The utilization of a data pump in the fax and modem relays to demodulate and remodulate data signals for transmission across a packet based network provides considerable bandwidth savings. First, only the underlying unmodulated data signals are transmitted across the packet based network. Second, data transmission rates of digital signals across the packet based network, typically 64 kbps is greater than the maximum rate available (typically 33,600 bps) for communication over a circuit switched network.

Telephone line data pumps operating in accordance with ITU V series recommendations for transmission rates of 2400 bps or more typically utilize quadrature amplitude modulation (QAM). A typical QAM data pump transmitter 600 is shown schematically in FIG. 30. The transmitter input is a serial binary data stream d_n arriving at a rate of R_d bps. A serial to parallel converter 602 groups the input bits into J-bit binary words. A constellation mapper 604 maps each J-bit binary word to a channel symbol from a 2^J element alphabet resulting in a channel symbol rate of $f_s=R_d/J$ baud. The alphabet consists of a pair of real numbers representing points in a two-dimensional space, called the signal constellation. Customarily the signal constellation can be thought of as a complex plane so that the channel symbol sequence may be represented as a sequence of complex numbers $c_n = a_n + jb_n$. Typically the real part a_n is called the in-phase or I component and the imaginary b_n is called the quadrature or Q component. A nonlinear encoder 605 may be used to expand the constellation points in order to combat the negative effects of companding in accordance with ITU-T G.711 standard. The I & Q components may be modulated by impulse modulators 606 and 608 respectively and filtered by transmit shaping filters 610 and 612 each with impulse response $g_T(t)$. The outputs of the shaping filters 610 and 612 are called in-phase 610(a) and quadrature 612(a) components of the continuous-time transmitted signal.

The shaping filters 610 and 612 are typically lowpass filters approximating the raised cosine or square root of raised cosine response, having a cutoff frequency on the order of at least about $f_s/2$. The outputs 610(a) and 612(a) of the lowpass filters 610 and 612 respectively are lowpass signals with a frequency domain extending down to approximately zero hertz. A local oscillator 614 generates quadrature carriers $\cos(\omega_c t)$ 614(a) and $\sin(\omega_c t)$ 614(b). Multipliers 616 and 618 multiply the filter outputs 610(a) and 612(a) by quadrature carriers $\cos(\omega_c t)$ and $\sin(\omega_c t)$ respectively to amplitude modulate the in-phase and quadrature signals up to the passband of a bandpass channel. The modulated output signals 616(a) and 618(a) are then subtracted in a difference operator 620 to form a transmit output signal 622. The carrier frequency should be greater than the shaping filter cutoff frequency to prevent spectral fold-over.

37367/CAG/B600

A data pump receiver 630 is shown schematically in FIG. 31. The data pump receiver 630 is generally configured to process a received signal 630(a) distorted by the non-ideal frequency response of the channel and additive noise in a transmit data pump (not shown) in the local modem. An analog to digital converter (A/D) 631 converts the received signal 630(a) from an analog to a digital format. The A/D converter 631 samples the received signal 630(a) at a rate of $f_0=1/T_0 = n_0/T$ which is n_0 times the symbol rate $f_s=1/T$ and is at least twice the highest frequency component of the received signal 630(a) to satisfy nyquist sampling theory.

An echo canceller 634 substantially removes the line echos on the received signal 630(a). Echo cancellation permits a modem to operate in a full duplex transmission mode on a two-line circuit, such as a PSTN. With echo cancellation, a modem can establish two high-speed channels in opposite directions. Through the use of digital-signal-processing circuitry, the modem's receiver can use the shape of the modem's transmitter signal to cancel out the effect of its own transmitted signal by subtracting reference signal and the receive signal 630(a) in a difference operator 633.

Multiplier 636 scales the amplitude of echo cancelled signal 633(a). A power estimator 637 estimates the power level of the gain adjusted signal 636(a). Automatic gain control logic 638 compares the estimated power level to a set of predetermined thresholds and inputs a scaling factor into the multiplier 636 that adjusts the amplitude of the echo canceled signal 633(a) to a level that is within the desired amplitude range. A carrier detector 642 processes the output of a digital resampler 640 to determine when a data signal is actually present at the input to receiver 630. Many of the receiver functions are preferably not invoked until an input signal is detected.

A timing recovery system 644 synchronizes the transmit clock of the remote data pump transmitter (not shown) and the receiver clock. The timing recovery system 644 extracts timing information from the received signal, and adjusts the digital resampler 640 to ensure that the frequency and phase of the transmit clock and receiver clock are synchronized. A phase splitting fractionally spaced equalizer (PSFSE) 646 filters the received signal at the symbol rate. The PSFSE 646 compensates for the amplitude response and envelope delay of the channel so as to minimize inter-symbol interference in the received signal. The frequency response of a typical channel is inexact so that an adaptive filter is preferable. The PSFSE 646 is preferably an adaptive FIR filter that operates on data signal samples spaced by T/n_0 and generates digital signal output samples spaced by the period T. In the described exemplary embodiment $n_0=3$.

The PSFSE 646 outputs a complex signal which multiplier 650 multiplies by a locally generated carrier reference 652 to demodulate the PSFSE output to the baseband signal 650(a). The received signal 630(a) is typically encoded with a non-linear operation so as to reduce the

37367/CAG/B600

quantization noise introduced by companding in accordance with ITU-T G.711. The baseband signal 650(a) is therefore processed by a non-linear decoder 654 which reverses the non-linear encoding or warping. The gain of the baseband signal will typically vary upon transition from a training phase to a data phase because modem manufacturers utilize different methods to compute a scale factor. The problem that arises is that digital modulation techniques such as quadrature amplitude modulation (QAM) and pulse amplitude modulation (PAM) rely on precise gain (or scaling) in order to achieve satisfactory performance. Therefore, a scaling error compensator 656 adjusts the gain of the receiver to compensate for variations in scaling. Further, a slicer 658 then quantizes the scaled baseband symbols to the nearest ideal constellation points, which are the estimates of the symbols from the remote data pump transmitter (not shown). A decoder 659 converts the output of slicer 658 into a digital binary stream.

During data pump training, known transmitted training sequences are transmitted by a data pump transmitter in accordance with the applicable ITU-T standard. An ideal reference generator 660, generates a local replica of the constellation point 660(a). During the training phase a switch 661 is toggled to connect the output 660(a) of the ideal reference generator 660 to a difference operator 662 that generates a baseband error signal 662(a) by subtracting the ideal constellation sequence 660(a) and the baseband equalizer output signal 650(a). A carrier phase generator 664 uses the baseband error signal 662(a) and the baseband equalizer output signal 650(a) to synchronize local carrier reference 666 with the carrier of the received signal 630(a) During the data phase the switch 661 connects the output 658(a) of the slicer to the input of difference operator 662 that generates a baseband error signal 662(a) in the data phase by subtracting the estimated symbol output by the slicer 658 and the baseband equalizer output signal 650(a). It will be appreciated by one of skill that the described receiver is one of several approaches. Alternate approaches in accordance with ITU-T recommendations may be readily substituted for the described data pump. Accordingly, the described exemplary embodiment of the data pump is by way of example only and not by way of limitation.

a. Timing Recovery System

Timing recovery refers to the process in a synchronous communication system whereby timing information is extracted from the data being received. In the context of a modem connection in accordance with an exemplary embodiment of the present invention, each modem is coupled to a signal processing system, which for the purposes of explanation is operating in a network gateway, either directly or through a PSTN line. In operation, each modem establishes a modem connection with its respective network gateway, at which point, the modems begin relaying data signals across a packet based network. The problem that arises is that the clock frequencies of the modems are not identical to the clock frequencies of the data pumps operating

37367/CAG/B600

in their respective network gateways. By design, the data pump receiver in the network gateway should sample a received signal of symbols in synchronism with the transmitter clock of the modem connected locally to that gateway in order to properly demodulate the transmitted signal.

A timing recovery system can be used for this purpose. Although the timing recovery system is described in the context of a data pump within a signal processing system with the packet data modem exchange invoked, those skilled in the art will appreciate that the timing recovery system is likewise suitable for various other applications in various other telephony and telecommunications applications, including fax data pumps. Accordingly, the described exemplary embodiment of the timing recovery system in a signal processing system is by way of example only and not by way of limitation.

A block diagram of a timing recovery system is shown in FIG. 32. In the described exemplary embodiment, the digital resampler 640 resamples the gain adjusted signal 636(a) output by the AGC (see FIG. 31). A timing error estimator 670 provides an indication of whether the local timing or clock of the data pump receiver is leading or lagging the timing or clock of the data pump transmitter in the local modem. As is known in the art, the timing error estimator 670 may be implemented by a variety of techniques including that proposed by Godard. The A/D converter 631 of the data pump receiver (see FIG. 31) samples the received signal 630(a) at a rate of fo which is an integer multiple of the symbol rate fs=1/T and is at least twice the highest frequency component of the received signal 630(a) to satisfy nyquist sampling theory. The samples are applied to an upper bandpass filter 672 and a lower bandpass filter 674. The upper bandpass filter 672 is tuned to the upper bandedge frequency fu = fc + 0.5fs and the lower bandpass filter 674 is tuned to the lower bandedge frequency fl = fc - 0.5fs where fc is the carrier frequency of the QAM signal. The bandwidth of the filters 672 and 674 should be reasonably narrow, preferably on the order of 100 Hz for a fs = 2400 baud modem. Conjugate logic 676 takes the complex conjugate of complex output of the lower bandpass filter. Multiplier 678 multiplies the complex output of the upper bandpass filter 672(a) by the complex conjugate of the lower bandpass filter to form a cross-correlation between the output of the two filters (672 and 674). The real part of the correlated symbol is discarded by processing logic 680, and a sampler 681samples the imaginary part of the resulting cross-correlation at the symbol rate to provide an indication of whether the timing phase error is leading or lagging.

In operation, a transmitted signal from a remote data pump transmitter (not shown) g(t) is made to correspond to each data character. The signal element has a bandwidth approximately equal to the signaling rate fs. The modulation used to transmit this signal element consists of multiplying the signal by a sinusoidal carrier of frequency fc which causes the spectrum to be translated to a band around frequency fc. Thus, the corresponding spectrum is bounded by

frequencies f1 = fc - 0.5fs and f2 = fc + 0.5fs, which are known as the bandedge frequencies. Reference for more detailed information may be made to "Principles of Data Communication" by R. W. Lucky, J. Salz and E. J. Weldon, Jr., McGraw-Hill Book Company, pages 50-51.

In practice it has been found that additional filtering is required to reduce symbol clock jitter, particularly when the signal constellation contains many points. Conventionally a loop filter 682 filters the timing recovery signal to reduce the symbol clock jitter. Traditionally the loop filter 682 is a second order infinite impulse response (IIR) type filter, whereby the second order portion tracks the offset in clock frequency and the first order portion tracks the offset in phase. The output of the loop filter drives clock phase adjuster 684. The clock phase adjuster controls the digital sampling rate of digital resampler 640 so as to sample the received symbols in synchronism with the transmitter clock of the modem connected locally to that gateway. Typically, the clock phase adjuster 684 utilizes a poly-phase interpolation algorithm to digitally adjust the timing phase. The timing recovery system may be implemented in either analog or digital form. Although digital implementations are more prevalent in current modem design an analog embodiment may be realized by replacing the clock phase adjuster with a VCO.

The loop filter 682 is typically implemented as shown in FIG. 33. The first order portion of the filter controls the adjustments made to the phase of the clock (not shown) A multiplier 688 applies a first order adjustment constant α to advance or retard the clock phase adjustment. Typically the constant α is empirically derived via computer simulation or a series of simple experiments with a telephone network simulator. Generally α is dependent upon the gain and the bandwidth of the upper and lower filters in the timing error estimator, and is generally optimized to reduce symbol clock jitter and control the speed at which the phase is adjusted. The structure of the loop filter 682 may include a second order component 690 that estimates the offset in clock frequency. The second order portion utilizes an accumulator 692 in a feedback loop to accumulate the timing error estimates. A multiplier 694 is used to scale the accumulated timing error estimate by a constant β. Typically, the constant β is empirically derived based on the amount of feedback that will cause the system to remain stable. Summer 695 sums the scaled accumulated frequency adjustment 694(a) with the scaled phase adjustment 688(a). A disadvantage of conventional designs which include a second order component 690 in the loop filter 682 is that such second order components 690 are prone to instability with large constellation modulations under certain channel conditions.

An alternative digital implementation eliminates the loop filter. Referring to FIG. 34 a hard limiter 695 and a random walk filter 696 are coupled to the output of the timing error estimator 680 to reduce timing jitter. The hard limiter 695 provides a simple automatic gain control action that keeps the loop gain constant independent of the amplitude level of the input

37367/CAG/B600

signal. The hard limiter 695 assures that timing adjustments are proportional to the timing of the data pump transmitter of the local modem and not the amplitude of the received signal. The random walk filter 696 reduces the timing jitter induced into the system as disclosed in "Communication System Design Using DSP Algorithms", S. Tretter, p. 132, Plenum Press, NY., 1995, the contents of which is hereby incorporated by reference as through set forth in full herein. The random walk filter 696 acts as an accumulator, summing a random number of adjustments over time. The random walk filter 696 is reset when the accumulated value exceeds a positive or negative threshold. Typically, the sampling phase is not adjusted so long as the accumulator output remains between the thresholds, thereby substantially reducing or eliminating incremental positive adjustments followed by negative adjustments that otherwise tend to not accumulate.

Referring to FIG. 35 in an exemplary embodiment of the present invention, the multiplier 688 applies the first order adjustment constant α to the output of the random walk filter to advance or retard the estimated clock phase adjustment. In addition, a timing frequency offset compensator 697 is coupled to the timing recovery system via switches 698 and 699 to preferably provide a fixed dc component to compensate for clock frequency offset present in the received signal. The exemplary timing frequency offset compensator preferably operates in phases. A frequency offset estimator 700 computes the total frequency offset to apply during an estimation phase and incremental logic 701, incrementally applies the offset estimate in linear steps during the application phase. Switch control logic 702 controls the toggling of switches 698 and 699 during the estimation and application phases of compensation adjustment. Unlike the second order component 690 of the conventional timing recovery loop filter disclosed in FIG. 33, the described exemplary timing frequency offset compensator 697 is an open loop design such that the second order compensation is fixed during steady state. Therefore, switches 698 and 699 work in opposite cooperation when the timing compensation is being estimated and when it is being applied.

During the estimation phase, switch control logic 702 closes switch 698 thereby coupling the timing frequency offset compensator 697 to the output of the random walk filter 696, and opens switch 699 so that timing adjustments are not applied during the estimation phase. The frequency offset estimator 700 computes the timing frequency offset during the estimation phase over K symbols in accordance with the block diagram shown in FIG. 36. An accumulator 703 accumulates the frequency offset estimates over K symbols. A multiplier 704 is used to average the accumulated offset estimate by applying a constant γ/K. Typically the constant γ is empirically derived and is preferably in the range of about 0.5-2. Preferably K is as large as possible to improve the accuracy of the average. K is typically greater than about 500 symbols and less than the recommended training sequence length for the modem in question. In the exemplary embodiment the first order adjustment constant α is preferably in the range of about

37367/CAG/B600

100-300 part per million (ppm). The timing frequency offset is preferably estimated during the timing training phase (timing tone) and equalizer training phase based on the accumulated adjustments made to the clock phase adjuster 684 over a period of time.

During steady state operation when the timing adjustments are applied, switch control logic 702 opens switch 698 decoupling the timing frequency offset compensator 697 from the output of the random walk filter, and closes switch 699 so that timing adjustments are applied by summer 705. After K symbols of a symbol period have elapsed and the frequency offset compensation is computed, the incremental logic 701 preferably applies the timing frequency offset estimate in incremental linear steps over a period of time to avoid large sudden adjustments which may throw the feedback loop out of lock. This is the transient phase. The length of time over which the frequency offset compensation is incrementally applied is empirically derived, and is preferably in the range of about 200-800 symbols. After the incremental logic 701 has incrementally applied the total timing frequency offset estimate computed during the estimate phase, a steady state phase begins where the compensation is fixed. Relative to conventional second order loop filters, the described exemplary embodiment provides improved stability and robustness.

b. Multipass Training

Data pump training refers to the process by which training sequences are utilized to train various adaptive elements within a data pump receiver. During data pump training, known transmitted training sequences are transmitted by a data pump transmitter in accordance with the applicable ITU-T standard. In the context of a modem connection in accordance with an exemplary embodiment of the present invention, the modems (see FIG. 29) are coupled to a signal processing system, which for the purposes of explanation is operating in a network gateway, either directly or through a PSTN line. In operation, the receive data pump operating in each network gateway of the described exemplary embodiment utilizes PSFSE architecture. The PSFSE architecture has numerous advantages over other architectures when receiving QAM signals. However, the PSFSE architecture has a slow convergence rate when employing the least mean square (LMS) stochastic gradient algorithm. This slow convergence rate typically prevents the use of PSFSE architecture in modems that employ relatively short training sequences in accordance with common standards such as V.29. Because of the slow convergence rate, the described exemplary embodiment re-processes blocks of training samples multiple times (multi-pass training).

Although the method of performing multi-pass training is described in the context of a signal processing system with the packet data exchange invoked, those skilled in the art will

37367/CAG/B600

appreciate that multi-pass training is likewise suitable for various other telephony and telecommunications applications. Accordingly, the described exemplary method for multi-pass training in a signal processing system is by way of example only and not by way of limitation.

In an exemplary embodiment the data pump receiver operating in the network gateway stores the received QAM samples of the modem's training sequence in a buffer until N symbols have been received. The PSFSE is then adapted sequentially over these N symbols using a LMS algorithm to provide a coarse convergence of the PSFSE. The coarsely converged PSFSE (i.e. with updated values for the equalizer taps) returns to the start of the same block of training samples and adapts a second time. This process is repeated M times over each block of training samples. Each of the M iterations provides a more precise or finer convergence until the PSFSE is completely converged.

c. Scaling Error Compensator

Scaling error compensation refers to the process by which the gain of a data pump receiver (fax or modem) is adjusted to compensate for variations in transmission channel conditions. In the context of a modem connection in accordance with an exemplary embodiment of the present invention, each modem is coupled to a signal processing system, which for the purposes of explanation is operating in a network gateway, either directly or through a PSTN line. In operation, each modem communicates with its respective network gateway using digital modulation techniques. The problem that arises is that digital modulation techniques such as QAM and pulse amplitude modulation (PAM) rely on precise gain (or scaling) in order to achieve satisfactory performance. In addition, transmission in accordance with the V.34 recommendations typically includes a training phase and a data phase whereby a much smaller constellation size is used during the training phase relative to that used in the data phase. The V.34 recommendation, requires scaling to be applied when switching from the smaller constellation during the training phase into the larger constellation during the data phase.

The scaling factor can be precisely computed by theoretical analysis, however, different manufacturers of V.34 systems (modems) tend to use slightly different scaling factors. Scaling factor variation (or error) from the predicted value may degrade performance until the PSFSE compensates for the variation in scaling factor. Variation in gain due to transmission channel conditions is compensated by an initial gain estimation algorithm (typically consisting of a simple signal power measurement during a particular signaling phase) and an adaptive equalizer during the training phase. However, since a PSFSE is preferably configured to adapt very slowly during the data phase, there may be a significant number of data bits received in error before the PSFSE has sufficient time to adapt to the scaling error.

37367/CAG/B600

It is, therefore, desirable to quickly reduce the scaling error and hence minimize the number of potential erred bits. A scaling factor compensator can be used for this purpose. Although the scaling factor compensator is described in the context of a signal processing system with the packet data modem exchange invoked, those skilled in the art will appreciate that the preferred scaling factor compensator is likewise suitable for various other telephony and telecommunications applications. Accordingly, the described exemplary embodiment of the scaling factor compensator in a signal processing system is by way of example only and not by way of limitation.

FIG. 37 shows a block diagram of an exemplary embodiment of the scaling error compensator in a data pump receiver 630 (see FIG. 31). In an exemplary embodiment, scaling error compensator 708 computes the gain adjustment of the data pump receiver. Multiplier 710 adjusts a nominal scaling factor 712 (the scaling error computed by the data pump manufacturer) by the gain adjustment as computed by the scaling error compensator 708. The combined scale factor 710(a) is applied to the incoming symbols by multiplier 714. A slicer 716 quantizes the scaled baseband symbols to the nearest ideal constellation points, which are the estimates of the symbols from the remote data pump transmitter.

The scaling error compensator 708 preferably includes a divider 718 which estimates the gain adjustment of the data pump receiver by dividing the expected magnitude of the received symbol 716(a) by the actual magnitude of the received symbol 716(b). In the described exemplary embodiment the magnitude is defined as the sum of squares between real and imaginary parts of the complex symbol. The expected magnitude of the received symbol is the output 716(a) of the slicer 716 (i.e. the symbol quantized to the nearest ideal constellation point) whereas the magnitude of the actual received symbol is the input 716(b) to the slicer 716. In the case where a Viterbi decoder performs the error-correction of the received, noise-disturbed signal (as for V.34), the output of the slicer may be replaced by the first level decision of the Viterbi decoder.

The statistical nature of noise is such that large spikes in the amplitude of the received signal will occasionally occur. A large spike in the amplitude of the received signal may result in an erroneously large estimate of the gain adjustment of the data pump receiver. Typically, scaling is applied in a one to one ratio with the estimate of the gain adjustment, so that large scaling factors may be erroneously applied when large amplitude noise spikes are received. To minimize the impact of large amplitude spikes and improve the accuracy of the system, the described exemplary scaling error compensator 708 further includes a non-linear filter in the form of a hard-limiter 720 which is applied to each estimate 718(a). The hard limiter 720 limits the maximum adjustment of the scaling value. The hard limiter 720 provides a simple automatic

37367/CAG/B600

control action that keeps the loop gain constant independent of the amplitude of the input signal so as to minimize the negative effects of large amplitude noise spikes. In addition, averaging logic 722 computes the average gain adjustment estimate over a number (N) of symbols in the data phase prior to adjusting the nominal scale factor 710. As will be appreciated by those of skill in the art, other non-linear filtering algorithms may also be used in place of the hard-limiter.

Alternatively, the accuracy of the scaling error compensation may be further improved by estimating the averaged scaling adjustment twice and applying that estimate in two steps. A large hard limit value (typically 1 +/- 0.25) is used to compute the first average scaling adjustment. The initial prediction provides an estimate of the average value of the amplitude of the received symbols. The unpredictable nature of the amplitude of the received signal requires the use of a large initial hard limit value to ensure that the true scaling error is included in the initial estimate of the average scaling adjustment. The estimate of the average value of the amplitude of the received symbols is used to calibrate the limits of the scaling adjustment. The average scaling adjustment is then estimated a second time using a lower hard limit value and then applied to the nominal scale factor 712 by multiplier 710.

In most modem specifications, such as the V.34 standards, there is a defined signaling period (B1 for V.34) after transition into data phase where the data phase constellation is transmitted with signaling information to flush the receiver pipeline (i.e. Viterbi decoder etc.) prior to the transmission of actual data. In an exemplary embodiment this signaling period may be used to make the scaling adjustment such that any scaling error is compensated for prior to actual transfer of data.

d. Non-Linear Decoder

In the context of a modem connection in accordance with an exemplary embodiment of the present invention, each modem is coupled to a signal processing system, which for the purposes of explanation is operating in a network gateway, either directly or through a PSTN line. In operation, each modem communicates with its respective network gateway using digital modulation techniques. The international telecommunications union (ITU) has promulgated standards for the encoding and decoding of digital data in ITU-T Recommendation G.711 (ref. G.711) which is incorporated herein by reference as if set forth in full. The encoding standard specifies that a nonlinear operation (companding) be performed on the analog data signal prior to quantization into seven bits plus a sign bit. The companding operation is a monatomic invertable function which reduces the higher signal levels. At the decoder, the inverse operation (expanding) is done prior to analog reconstruction. The companding / expanding operation quantizes the higher signal values more coarsely. The companding / expanding operation, is

37367/CAG/B600

suitable for the transmission of voice signals but introduces quantization noise on data modem signals. The quantization error (noise) is greater for the outer signal levels than the inner signal levels.

The ITU-T Recommendation V.34 describes a mechanism whereby (ref. V.34) the uniform signal is first expanded (ref. BETTS) to space the outer points farther apart than the inner points before G.711 encoding and transmission over the PCM link. At the receiver, the inverse operation is applied after G.711 decoding. The V.34 recommended expansion / inverse operation yields a more uniform signal to noise ratio over the signal amplitude. However, the inverse operation specified in the ITU-T Recommendation V.34 requires a complex receiver calculation. The calculation is computationally intensive, typically requiring numerous machine cycles to implement.

It is, therefore, desirable to reduce the number of machine cycles required to compute the inverse to within an acceptable error level. A simplified nonlinear decoder can be used for this purpose. Although the nonlinear decoder is described in the context of a signal processing system with the packet data modem exchange invoked, those skilled in the art will appreciate that the nonlinear decoder is likewise suitable for various other telephony and telecommunications application. Accordingly, the described exemplary embodiment of the nonlinear decoder in a signal processing system is by way of example only and not by way of limitation.

Conventionally, iteration algorithms have been used to compute the inverse of the G.711 nonlinear warping function. Typically, iteration algorithms generate an initial estimate of the input to the nonlinear function and then compute the output. The iteration algorithm compares the output to a reference value and adjusts the input to the nonlinear function. A commonly used adjustment is the successive approximation wherein the difference between the output and the reference function is added to the input. However, when using the successive approximation technique, up to ten iterations may be required to adjust the estimated input of the nonlinear warping function to an acceptable error level, so that the nonlinear warping function must be evaluated ten times. The successive approximation technique is computationally intensive, requiring significant machine cycles to converge to an acceptable approximation of the inverse of the nonlinear warping function. Alternatively, a more complex warping function is a linear Newton Rhapson iteration. Typically the Newton Rhapson algorithm requires three evaluations to converge to an acceptable error level. However, the inner computations for the Newton Rhapson algorithm are more complex than those required for the successive approximation technique. The Newton Rhapson algorithm utilizes a computationally intensive iteration loop wherein the derivative of the nonlinear warping function is computed for each approximation

37367/CAG/B600

iteration, so that significant machine cycles are required to conventionally execute the Newton Rhapson algorithm.

An exemplary embodiment of the present invention modifies the successive approximation iteration. A presently preferred algorithm computes an approximation to the derivative of the nonlinear warping function once before the iteration loop is executed and uses the approximation as a scale factor during the successive approximation iterations. The described exemplary embodiment converges to the same acceptable error level as the more complex conventional Newton-Rhapson algorithm in four iterations. The described exemplary embodiment further improves the computational efficiency by utilizing a simplified approximation of the derivative of the nonlinear warping function.

In operation, development of the described exemplary embodiment proceeds as follows with a warping function defined as:

$$w(v) = \frac{\Theta(v)}{6} + \frac{\Theta(v)^2}{120}$$

the V.34 nonlinear decoder can be written as

$$Y = X(1 + w(\|X\|^2))$$

taking the square of the magnitude of both sides yields,

$$Y^2 = |X|^2 (1 + w(\|X\|^2))^2$$

The encoder notation can then be simplified with the following substitutions

$$Y_r = \|Y\|^2, X_r = \|X\|^2$$

and write the V.34 nonlinear encoder equation in the cannonical form G(x)=0.

$$X_r(1 + w(X_r))^2 - Y_r = 0$$

The Newton-Rhapson iteration is a numerical method to determine X that results in an iteration of the form:

$$X_{n+1} = X_n - \frac{G(X_n)}{G'(X_n)}$$

37367/CAG/B600

where G′ is the derivative and the substitution iteration results when G′ is set equal to one.

The computational complexity of the Newton-Rhapson algorithm is thus paced by the derivation of the derivative G′, which conventionally is related to X_r so that the mathematical instructions saved by performing fewer iterations are offset by the instructions required to calculate the derivative and perform the divide. Therefore, it would be desirable to approximate the derivative G′ with a term that is the function of the input Y_r so that G(x) is a monotonic function and G′(x) can be expressed in terms of G(x). Advantageously, if the steps in the iteration are small, then G′(x) will not vary greatly and can be held constant over the iteration. A series of simple experiments yields the following approximation of G′(x) where α is an experimentally derived scaling factor.

$$G' = \frac{1+Yr}{\alpha}$$

The approximation for G′ converges to an acceptable error level in a minimum number of steps, typically one more iteration than the full linear Newton-Rhapson algorithm. A single divide before the iteration loop computes the quantity

$$\frac{1}{G'} = \frac{\alpha}{1+Yr}$$

The error term is multiplied by 1/G′ in the successive iteration loop. It will be appreciated by one of skill in the art that further improvements in the speed of convergence are possible with the "Generalized Newton-Rhapson" class of algorithms. However, the inner loop computations for this class of algorithm are quite complex.

Advantageously, the described exemplary embodiment does not expand the polynomial because the numeric quantization on a store in a sixteen bit machine may be quite significant for the higher order polynomial terms. The described exemplary embodiment organizes the inner loop computations to minimize the effects of truncation and the number of instructions required for execution. Typically the inner loop requires eighteen instructions and four iterations to converge to within two bits of the actual value which is within the computational roundoff noise of a sixteen bit machine.

D. Human Voice Detector

In a preferred embodiment of the present invention, a signal processing system is employed to interface telephony devices with packet based networks. Telephony devices include, by way of example, analog and digital phones, ethernet phones, Internet Protocol phones, fax

37367/CAG/B600

machines, data modems, cable voice modems, interactive voice response systems, PBXs, key systems, and any other conventional telephony devices known in the art. In the described exemplary embodiment the packet voice exchange is common to both the voice mode and the voiceband data mode. In the voiceband data mode, the network VHD invokes the packet voice exchange for transparently exchanging data without modification (other than packetization) between the telephony device or circuit switched network and the packet based network. This is typically used for the exchange of fax and modem data when bandwidth concerns are minimal as an alternative to demodulation and remodulation.

During the voiceband data mode, the human voice detector service is also invoked by the resource manager. The human voice detector monitors the signal from the near end telephony device for voice. The described exemplary human voice detector estimates pitch period of an incoming telephony signal and compares the pitch period of said telephony signal to a plurality of thresholds to identify active voice samples. This approach is substantially independent of the amplitude of the spoken utterance, so that whispered or shouted utterance may be accurately identified as active voice samples. In the event that voice is detected by the human voice detector, an event is forwarded to the resource manager which, in turn, causes the resource manager to terminate the human voice detector service and invoke the appropriate services for the voice mode (i.e., the call discriminator, the packet tone exchange, and the packet voice exchange).

Although a preferred embodiment is described in the context of a signal processing system for telephone communications across the packet based network, it will be appreciated by those skilled in the art that the voice detector is likewise suitable for various other telephony and telecommunications application. Accordingly, the described exemplary embodiment of the voice detector in a signal processing system is by way of example only and not by way of limitation.

There are a variety of encoding methods known for encoding voice. Most frequently, voice is modeled on a short-time basis as the response of a linear system excited by a periodic impulse train for voiced sounds or random noise for the unvoiced sounds. Conventional human voice detectors typically monitor the power level of the incoming signal to make a voice / machine decision. Typically, if the power level of the incoming signal is above a predetermined threshold, the sequence is typically declared voice. The performance of such conventional voice detectors may be degraded by the environment, in that a very soft spoken whispered utterance will have a very different power level from a loud shout. If the threshold is set at too low a level, noise will be declared voice, whereas if the threshold is set at too high a level a soft spoken voice segment will be incorrectly marked as inactive.

37367/CAG/B600

Alternatively, voice may generally be classified as voiced if a fundamental frequency is imported to the air stream by the vocal cords of the speaker. In such case, the frequency of a voice segment is typically highly periodic at around the pitch frequency. The determination as to whether a voice segment is voiced or unvoiced, and the estimation of the fundamental frequency can be obtained in a variety of ways known in the art such as pitch detection algorithms. In the described exemplary embodiment, the human voice detector calculates an autocorrelation function for the incoming signal. An autocorrelation function for a voice segment demonstrates local peaks with a periodicity in proportion to the pitch period. The human voice detector service utilizes this feature in conjunction with power measurements to distinguish voice signals from modem signals. It will be appreciated that other pitch detection algorithms known in the art can be used as well.

Referring to FIG. 38, in the described exemplary embodiment, a power estimator 730 estimates the power level of the incoming signal. Autocorrelation logic 732 computes an autocorrelation function for an input signal to assist in the voice/machine decision. Autocorrelation, as is known in the art, involves correlating a signal with itself. A correlation function shows how similar two signals are, and how long the signals remain similar when one is shifted with respect to the other. Periodic signals go in and out of phase as one is shifted with respect to the other, so that a periodic signal will show strong correlation at shifts where the peaks coincide. Thus, the autocorrelation of a periodic signal is itself a periodic signal, with a period equal to the period of the original signal.

The autocorrelation calculation computes the autocorrelation function over an interval of 360 samples with the following approach:

$$R[k] = \sum_{n=0}^{N-k-1} x[n]x[n+k]$$

where N=360, k=0,1,2...179.

A pitch tracker 734 estimates the period of the computed autocorrelation function. Framed based decision logic 736 analyzes the estimated power level 730a, the autocorrelation function 732a and the periodicity 734a of the incoming signal to execute a frame based voice/machine decision according to a variety of factors. For example, the energy of the input signal should be above a predetermined threshold level, preferably in the range of about -45 to -55 dBm, before the frame based decision logic 736 declares the signal to be voice. In addition, the typical pitch period of a voice segment should be in the range of about 60-400 Hz, so that the autocorrelation function should preferably be periodic with a period in the range of about 60-400 Hz before the frame based decision logic 736 declares a signal as active or containing voice.

37367/CAG/B600

The amplitude of the autocorrelation function is a maximum for R[0], i.e. when the signal is not shifted relative to itself. Also, for a periodic voice signal, the amplitude of the autocorrelation function with a one period shift (i.e. R[pitch period]) should preferably be in the range of about 0.25-0.40 of the amplitude of the autocorrelation function with no shift (i.e. R[0]). Similarly, modem signaling may involve certain DTMF or MF tones, in this case the signals are highly correlated, so that if the largest peak in the amplitude of the autocorrelation function after R[0] is relatively close in magnitude to R[0], preferably in the range of about 0.75-0.90 R[0], the frame based decision logic 736 declares the sequence as inactive or not containing voice.

Once a decision is made on the current frame as to voice or machine, final decision logic 738 compares the current frame decision with the two adjacent frame decisions. This check is known as backtracking. If a decision conflicts with both adjacent decisions it is flipped, i.e. voice decision turned to machine and vice versa.

6) HPNA VoIP Timing Synch Circuit

HPNA VoIP Timing Synch Circuit

This document presents a solution to the problem of synchronization of clocks between the Cable Modem (CM) and the handset in a VoIP network that includes an HPNA LAN as the link between the handset and the CM. The clock in the cable modem is used to synchronize transmissions of upstream packets to the DOCSIS MAC timing. Upstream transmission times are generally dictated by the DOCSIS head end equipment. In addition, for~~For~~ synchronous traffic flows, such as VoIP, the periodicity of the transmission of packets of the flow is directly related to the upstream clock. Furthermore, the data samples in the packets are acquired at a rate which is a derivative of the system master clock. Because of ~~this~~these timing relationships, the cable modem clock must be synchronized to the clock in the cable modem head end.

At the VoIP handset, the local clock is used to sample the analog voice channel. This local clock must be related to the DOCSIS head end clock for proper operation to occur.

1 The Need for Synchronization

Synchronization between clocks in VoIP handsets and CMs is necessary for two reasons:

1. The sample rate of the analog voice signal at the handset must match a standard 8kHz value that is established for the entire voice transmission path in order to avoid frame slips (lost samples or sample gaps) which compromise the quality of voice traffic and significantly reduce the throughput of voice-band data flows.
2. The framing of samples into an RTP voice packet must occur synchronously to the arrival of an upstream grant at the DOCSIS MAC in order to minimize the latency of the upstream path.

The SNR of the coded voice signal that traverses the PSTN must meet the requirements of ITU-T recommendation G.712, which specifies an SNR of 35.5dB for most input levels. Variation in the A/D sample clock from a nominal 8kHz frequency can be modeled as noise in the coded signal, and therefore, a poorly tracking sample clock in the handset can cause the handset to fall out of compliance with ITU-T G.712. The performance limits of G.712 translate directly into the jitter performance objective for the timing synchronization circuit of the HPNA VoIP system.

A voice sample loss rate of 0.25 samples lost per minute must be maintained to support a toll-quality VoIP call. This requirement translates into a long-term average tracking error of 0.52ppm between the handset and the CM.

The overall latency that can be experienced by a real-time interactive voice call before user-reported degradation of call quality occurs has been determined, through experimentation, to be no more than 150msec according to ITU-T recommendation G.114. Therefore, the one-way latency limit of 150msec from ITU-T G.114 sets the performance goal for the latency requirement to be met by the HPNA VoIP system. The largest potential customer of the systems to be built using the HPNA LAN for VoIP traffic has stated their desire for the final system to be capable of meeting the G.114 goal.

2 The means for time synchronization

Both the CM and the handset will contain a local reference clock for the HPNA LAN. The two clocks must share a common value and must be running at the same rate, averaged over time, with a maximum instantaneous error not to exceed TBD, which matches the DOCSIS requirements.

Several mechanisms have been explored in order to solve the synchronization problem. Among them:

A software mechanism for determining the timestamp at a remote location and correlating that time to the local time, using round trip estimation to determine the correction for queuing delay at each end. E.g. Network Time Protocol.

A relative adjustment mechanism that sends only corrective indications between the timing master and the timing slave.

Both of these methods lack the ability to discriminate between timing errors that are due to frequency drift at the slave and errors that are due to inaccuracies in determining the exact reference time. It is not well known if the inaccuracy of determining the reference time might create frequent and wide swings in the local reference clock, resulting in widely varying sample intervals over relatively short periods of time, or worse, resulting in unstable clock behavior and frame slips. If wide or sudden variations in reference time information is expected, then a reduction in tracking loop gain might solve the problem, but such a reduction might place the tracking ability below the level where actual frequency drift can be tracked well enough to meet the performance criteria for VoIP! Perhaps the most compelling argument against a soft method of time determination and tracking is the one that suggests that while the frequencies in question may remain relatively stable over the periods of interest, the reference time establishment methodology (round trip time measurements) may not be very stable over short periods of time. Changing traffic patterns may produce sudden and persistent asymmetries in the two legs of the round trip, resulting in a sudden change in the timestamp estimation error. Without distinguishing the reference time estimation error from the frequency drift error, it could be the case that the DPLL inappropriately uses frequency corrections to adjust for these sudden phase shifts. The sampling frequency could then be enough out of step with the CM as to cause frame slips over relatively short periods of time. Voice-band data might suffer throughput degradation from the relative sampling time errors and voice traffic itself might suffer from harmonic distortions. The SNR requirements of ITU-T G.712 might not be met.

In any case, any of these methods ultimately require the implementation of a local clock generation circuit with a tracking function in order to create a clock source for the A/D circuit at the handset. Given that the need for a tracking function is required, there is only a little extra work needed to include a more formal mechanism for delivering precise reference time information that does not confuse frequency drift with reference time estimation error.

2.1 A/D sample clock jitter

The cable modem products employ a DPLL to track the reference clock which is located in the cable modem head end equipment. The performance of the DPLL must be sufficient to meet the requirements for digitized voice transmission set forth in ITU-T recommendation G.712.

ITU-T recommendation G.712 gives an SNR of 35dB to be maintained for PCM signals. This value cannot be met with PCM μ-law encoding (beginning with 12-bit linear samples) in the presence of more than about −70dB noise. The analysis done for the voice over DOCSIS case, accounting for the A/D and D/A performance, suggests that the output clock used for generating the 8kHz A/D voice sampling clock should have a jitter of 5ns or less in order to meet these requirements. Any DPLL employed for clock tracking must be able to perform to this level if G.712 criteria are to be met.

Assuming that the highest sampled frequency in the voice band is 4kHz, then with 5ns of jitter, a sine wave of 4kHz experiences a maximum instantaneous amplitude error of:

$$20 * \log[\sin(5ns / 250\mu\sec * 2\pi) - \sin(0)] = -78dB$$

A jitter of 30ns produces an error of:

$$20 * \log[\sin(30ns / 250\mu\sec * 2\pi) - \sin(0)] = -62dB$$

The existing HPNA MAC includes a clock of 64MHz, which could produce a jitter of 15.7ns:

$$20 * \log[\sin(15.7ns / 250\mu\sec * 2\pi) - \sin(0)] = -68dB$$

One further point to note is that the CM device currently does not provide a straightforward means for determining grant arrival times to the MIPS core. This information is, however, available through a five-pin interface on the BCM3350 and its follow-ons. These facts point favorably in the direction of at least a partial hardware solution for collection and delivery of grant and reference timing information.

The general mechanism that should be used to maintain timer synchronization between the CM and the HPNA handset is very close to the method used by the CM and the head end equipment in the DOCSIS network – however, as much of the circuit as is possible has been moved to software. This minimizes the impact to the MAC design while maintaining some flexibility in the design that allows the synchronization mechanism to be fine-tuned outside of the silicon development schedule.

2.2 DOCSIS time & grant synchronization

The CM's DOCSIS clock maintains synchronization with the headend DOCSIS clock through the exchange of ranging messages and SYNC messages with the DOCSIS head end equipment. The timestamps in these messages are inserted and extracted as the messages leave or enter the DOCSIS MAC devices. The synchronization of the CM clock is maintained by a circuit within the DOCSIS MAC called the Timing Regeneration Circuit (TRC). The CM extracts the timestamp from the SYNC message as the bits are arriving off of the wire. This timestamp is passed to the TRC, where an immediate comparison to the local timestamp is made. Any difference is used to adjust a DPLL which controls the local clock frequency. A ranging message is used to determine the time-distance between the CM and the head end. The local clock is adjusted for this offset.

The local clock in the CM is used to time CM DOCSIS operations, such as upstream transmissions. But CM VoIP operations must also run synchronously to the DOCSIS head end clock, so the BCM3350 device includes two functions which allow for POTS/VoIP conversion devices (i.e. A/D and codec functions) to operate in synch with the DOCSIS clock.

The first VoIP support function of the BMC3350 is the export of a clock (TIC_CLK_OUT), which is a derivative of the local DOCSIS clock. TIC_CLK_OUT is used to drive the A/D sampling of the voice channel. This clock is used in order to insure that the sample rate of the A/D is locked in frequency to the DOCSIS clock. By doing this, the A/D sampling does not get ahead of or behind the DOCSIS grants – a situation which would result in lost samples or gaps in the stream of samples.

The second VoIP support function of the BMC3350 device is the export of a set of grant signals which indicate the arrival time of an upstream grant which corresponds to the desired framing interval of the collected voice samples. This grant signal indicates the framing boundary for a Voice over IP RTP data packet, which is a collection of A/D compressed and coded samples.

An equivalent of these two functions must be exported to the HPNA LAN-attached handsets, in order to allow the analog portion of the handset to maintain a proper sample rate and to allow the DSP to packetize a set of samples in a timely manner, to avoid additional path latency.

2.3 HPNA time & grant synchronization

The HPNA device does not need to duplicate the exact mechanism of the DOCSIS MAC device because the HPNA MAC at the CM has direct access to the TICK_CLK_OUT clock. Therefore, a subset of the DOCSIS synchronization mechanism is prescribed for the HPNA LAN MAC device.

In addition, the HPAN LAN MAC must mimic both the DOCSIS head end behavior and the DOCSIS CPE behavior. The HPNA LAN MAC device located at the CM will provide a timing reference to the HPNA LAN MAC devices located in handsets. The CM's HPNA MAC will mimic the functionality of the head end equipment with respect to clock sourcing. That is, there will be a master/slave relationship between HPNA MAC's in CMs and HPNA MACs in handsets – the master dictates the current time to the slaves.

This relationship only slightly complicates the HPNA MAC time synchronization solution, as the same circuit can easily be made to operate in either capacity.

The basic solution is similar to the DOCSIS MAC solution. A DPLL is incorporated within the HPNA MAC device. The DPLL is easily obtained as a complete circuit (Timing Regeneration Circuit) from the CM design team. In addition, the Smoothed TICK Clock Generator circuit is needed to produce the A/D sample clock at the handset side. Some minor modifications to the TRC are necessary.

In addition to the DPLL, the HPNA MAC needs to include a grant timing indication circuit. This circuit is basically a timestamp function that operates whenever a grant is signaled by the CM. In practice, it is simply a modification to the existing CM DPLL circuit.

A few registers are added to the HPNA MAC to support the TRC operation, and a few more for supporting the Grant Timing Indication circuit. These registers are fully described later in the document.

The final modification to the HPNA MAC is to include up to 6 new pins to provide an interface into the new circuits. In fact, the handset requires only 2 pins to support the needed synchronization function. The 6 pins is a maximum requirement for the timing master configuration. The timing slave needs only 2 pins. It has been suggested that the timing slave provide 3 pins as shown in the table. The pins employed for the master functions do not need to be shared with the pins that support the slave functions. The pins will operate differently depending upon whether the MAC is at the CM or at the handset. The pins provide the following functionality:

PIN NAME	CM-side Function (HPNA timing master)		Handset Function (HPNA timing slave)	
DPLL_REF_CLK	DPLL input clock	IN		
Grant[4]	Grant Present Indication	IN		
Grant[3]	Grant SID Value[3]	IN		
Grant[2]	Grant SID Value[2]	IN		
Grant[1]	Grant SID Value[1]	IN		
Grant[0]	Grant SID Value[0]	IN		
V_CLK_OUT			DPLL output clock	OUT
GPI[0]			Grant Present Indication[0]	OUT
GPI[1]			Grant Present Indication[1]	OUT

There is some unsettled discussion surrounding the question of whether or not additional Grant Present Indications are needed by the handset. That is, should the handset HPNA MAC be capable of providing grant indications for more than one VoIP connection?

Because the current Broadcom CM reference design utilizes the MSI mode of the HPNA MAC device, the 6 pins can be multiplexed with the upper AD pins of the PCI interface when in MSI mode. It is not expected that other CM designs which might employ the PCI bus would also include the GrantRcv and reference clock signals used by this interface. It is also not expected that PC-telephony applications need to be supported, therefore, the timing synchronization function will not be available in PCI mode.

One product requiring both the use of the PCI mode and the grant synchronization interface has been suggested. This product would be a PCI-based HPNA card for a PC, in which an RJ11 jack would be provided to allow for a single POTS line connection to the back of the PC. The card would serve a dual purpose of providing a data communications path for the PC while allowing the user to add a new VoIP line to his existing set of phone lines. This product would necessarily cost more than a stand-alone PCI data-only card, since it would have to include the A/D, DSP, memory and miscellaneous functions required to convert the POTS signal to HPNA. In any case, if the reality of this type of product is considered quite likely, then the PCI-based grant interface needs to be factored into the pin configuration of the PCI mode.

In any case, if the most likely PCI-based grant interface scenarios represent only handset applications, then only three pins are needed to supply a complete enough interface. It may be possible to reduce this to two pins, if the DPLL input clock can be obtained from an existing, internal HPNA MAC clock.

2.3.1 Time Synchronization

At the CM side, the HPNA MAC uses the CM's TICK_CLK_OUT signal as the reference input to the DPLL. Since this reference is already locked to the head-end's DOCSIS clock, no corrections are ever needed for the DPLL that operates in the HPNA MAC at the CM site – it too runs in synch with the DOCSIS clock. Note that no attempt is made to make the *value* of the CM HPNA MAC timer match the *value* of the DOCSIS MAC timer. This is not necessary. However, it will be necessary to match the timer value in the CM to the timer value in the handset.

The synchronized reference clock information needs to be transferred from the CM HPNA MAC to the HPNA handsets so that local sampling operations can maintain synchronization with the DOCSIS reference, and so that the handsets can frame their samples to align with Upstream Grant arrivals. The transfer of the CM HPNA MAC timestamp to the handset HPNA MAC timers is effected as follows:

Instead of transferring DOCSIS SYNC-like messages with timestamps inserted/extracted on the fly, the HPNA synchronization mechanism relies on an internal MAC indication of frame movement to latch the current time into a timestamp register. The value in the register is read and then delivered in a subsequent frame to the handset which uses it to adjust its clock.

The CM HPNA MAC device is set up (through a register bit) to be a timing master, such that only transmit activity is timestamped. Ideally, only frames marked with the Timestamp transmit descriptor bit will cause the HPNA MAC timestamp to be latched. Software in the CM reads the timestamp following the sending of a frame that had the Timestamp descriptor bit set to TRUE. Software then constructs a TIMESTAMP REPORT message containing the latched timestamp value and queues this frame for HPNA LAN delivery to the broadcast address. The queue latency is unknown and doesn't matter. The strict identity of the frame which generated the timestamping event is unknown and doesn't matter, although it is preferable to limit the frames which are timestamped. The mechanism chosen is to timestamp only TX frames that have the LTS descriptor bit set. To limit processing requirements at the receive end, a special message type, Timestamp Report Message (TRM), is defined. Only TRM will need to have timestamp information recorded and delivered from the timing master. Timing slaves will then be able to ignore receive timestamp information from all but TRM packets.

Meanwhile, at the handset, the receiver has been configured to act as a timing slave, such that only receive activity is timestamped. Each received frame triggers a timestamp to occur at the same relative position within a frame. There is a tradeoff wherein positioning the timestamp sample at an earlier location in the frame (up to and including the Type/Length field) yields a fixed offset from the beginning of the frame and results in the elimination of an offset correction. But the earlier timestamp allows less time for the handset's logic to read the latched timestamp before a new frame possibly overwrites the latched value. A preferred method causes the latched timestamp to be incorporated within the RX status word of each received frame, thereby eliminating any race condition. In any case, the timestamp for each received frame is stored in memory. Associated with each timestamp is a TRM sequence number. The receiver may eliminate all RX status word timestamps that do not correspond to TRM packets. What remains is a database of TRM sequence numbers and their corresponding RX timestamps.

When a TIMESTAMP REPORT message arrives, the handset searches its local database for the referenced sequence number and compares the received timestamp with the stored timestamp. The difference between the two values is used to determine the DPLL error. The handset performs a filtering function on the error, adds the DPLL bias value and then writes the resulting value into the NCO_INC register. In order to maximize the performance of the DPLL, it is recommended that TRM packets be sent in pairs. The rate of transmission is TBD, but suggested at about 1 pair per second.

From the DPLL, an output can be fed to the pin output that will drive the codec of the handset and ultimately, the A/D sampling circuit.

2.3.1.1 Initialization of handset timestamp value

Initialization of the handset timer is achieved by accepting two TIMESTAMP REPORT messages, the second one of which refers to the first. The reciever adopts the error indicated as an OFFSET value. This value is always added to received timestamps in order to calculate DPLL error. The DPLL counter is never modified. Since part of the DPLL loop is performed in software, the offset correction can easily be performed there.

2.3.2 Grant Synchronization

The CM HPNA clock must be sampled as DOCSIS upstream grants arrive. The grant arrival times will then be communicated to individual handsets through HPNA packets, in order to allow the assembly and queuing of RTP voice packets to be scheduled to insure that the packets will arrive at the CM just in time for the next upstream grant. Packet assembly overhead, queuing latency, transmission time, and CM packet processing time must be subtracted from the grant time in order to generate a packet assembly start time that insures that the packet meets the next upstream grant at the CM. The mechanics of this operation are as follows:

DOCSIS upstream grants are signaled by the BCM3350 through the GrantRcv[4:0] interface. GrantRcv[4] is used to indicate the arrival of a grant from the head end. GrantRcv[3:0] are used to signal the SID which corresponds to the current grant. Each SID corresponds to a particular connection flow, such as an individual call flow. The timing of the arrival of each grant needs to be communicated to the appropriate handset. In order to accomplish this, the 5 GrantRcv signals are fed to the CM HPNA MAC, and the HPNA MAC's internal timestamp value is latched whenever the GrantRcv[4] signal becomes active, provided that the GrantRcv[3:0] signals match the value set up in the tscSID register of the HPNA MAC. The MIPS core of the CM programs the tscSID register to match the SID corresponding to the call in progress for a given handset. Once the GrantRcv[4] timing is latched in the HPNA MAC, the MIPS core reads the latched timestamp and subtracts worst case queuing latency, transmission time, and CM packet processing time. It then sends a GRANT_TIMESTAMP message to the appropriate handset. A SID to MAC address mapping must exist at the CM in order to allow for proper grant timing signaling. This map is constructed and maintained by the MIPS core.

The handset receives the GRANT_TIMESTAMP message (an extended version of the TIMESTAMP REPORT message). The handset adds N*T time units (N= integer, T= RTP packet period) minus packet assembly processing latency to the timestamp from the message in order to calculate a time that is in the future. It then loads this time into the GRANT_TIME register so that the HPNA MAC can produce a grant-sync output to the codec at the appropriate time. When the TRC reaches GRANT_TIME, the GrantRcv[4] signal is asserted for one clock pulse duration and the GRANT_TIME register is automatically incremented by the value in the GRANT_PERIOD register. A register bit exists to disable the generation of grant pulses on GrantRcv[4].

We need a safety bit to indicate that the grant time has been indicated, in order to prevent the case of a grant time having been passed before it was programmed, and hence, no grant signals ever being generated? The safety bit would be a register bit that changes from a 0 to a 1 when the grant time is signaled on the output pin, and which can only be reset to 0 by software.

Note that the timing master must switch between transmit and grant-arrival timestamp latching operations. The implementation may include either one latch that is switchable between the two functions, or two latches to satisfy both requirements. The receive frame timestamp latching operation may share one of the latches mentioned, or it may be separate.

3 HPNA MAC changes

3.1 Pins

PIN NAME	CM-side Function (HPNA timing master)		Handset Function (HPNA timing slave)	
DPLL_REF_CLK	DPLL input clock	IN		
Grant[4]	Grant Present Indication	IN		
Grant[3]	Grant SID Value[3]	IN		
Grant[2]	Grant SID Value[2]	IN		
Grant[1]	Grant SID Value[1]	IN		
Grant[0]	Grant SID Value[0]	IN		
V_CLK_OUT			DPLL output clock	OUT
Frame[0]			Frame boundary marker[0]	OUT
Frame[1]			Frame boundary marker[1]	OUT

The device is either a timing master or a timing slave, but never both. Therefore, the maximum number of pins required for either mode is 6. This requirement is for the timing master, where the MSI mode is expected to be employed.

3.2 Registers:

Newly defined registers for the HPNA MAC. These registers did not come with the TRC circuit.

NCO_INC[15:0]	written with the filtered difference between slave and master time plus NCO bias value when tracking adjustments are being made to the DPLL
tscSID[3:0]	determines which Grant[4] input pulses will cause a timestamp latch event – latch events only occur when Grant[3:0] match tscSID[3:0] AND Grant[4] is asserted AND tMastertMaster is TRUE AND sGrant is true
GRANT_TIME[15:0]	contains a time that is to be matched against the slave time + offset_adjust. When a match occurs, Grant[4] output is asserted for one clock pulse and the value of GRANT_TIME is automatically incremented by the value of GRANT_PERIOD (multiple registers to support multiple channels?)
GRANT_PERIOD[15:0]	(fixed at 10msec, so not needed?)
TX_TIMESTAMP[31:0]	contains timestamp latched as a result of a transmit event (e.g. preamble transmitted AND TIMESTAMP bit of TX descriptor is TRUE?) (shared with GRANT TIMESTAMP register)
RX_TIMESTAMP[31:0]	contains timestamp latched as a result of a receive event (e.g. DA = BCAST?), the lower 16 bits of this value will be automatically stored in the RX status word
V_SCALE[7:0]	scaling value to be applied to the timestamp clock in order to produce the required A/D voice sampling clock
TS_SCALE[7:0]	scaling value to be applied to the NCO output clock in order to create a common Timestamp clock frequency

3.3 Misc. Register bits:

These register bits could go into existing registers if needed.

EN_REF_OUT	when set, this bit enables the V_CLK_OUT and Grant[4:3] output drive functions. This control bit only causes these pins to become outputs when the chip mode is MSI.
S_EXT_REF_CLK	when set, the TRC circuit input reference clock source is the DPLL_REF_CLK pin, when reset, the TRC input clock source is internal to the device
tMastertMaster	used to switch between latching timestamp on transmit signal instead of receive signal, . default value is tMastertMaster = TRUE
sGrant	used to switch between latching timestamp on Grant[4] signal instead of on transmit signal
GRANT_SIGNALED	needed to make sure that the Frame[0] signal was actually asserted – the slave controller may have set a GRANT_TIME that was not sufficiently far in the future, due to processing latency – if the GRANT_TIME value had already been passed when it was loaded, then no grant signals are being generated externally – this bit can be used to verify that the GRANT_TIME value has been reached (is this necessary? – our only timing problem would be the cycles between receiving the GRANT_TIMESTAMP message and calculating a future time, then loading the GRANT_TIME register...no queuing latency is involved) This bit is resetable by the host.
S_DPLL_OUT	when set, this bit causes the V_CLK_OUT mux to use the DPLL output clock directly, without passing through the two integer dividers.
S_NCO_TS	Used to select the NCO output, or the second integer divider output as the clock which drives the Timestamp counter. When this bit and S_REF_TS are both set to 1, then the NCO output clock is used to drive the timestamp counter. When this bit is set to zero and the S_REF_TS bit is set to one, then the second divider output is used to drive the timestamp counter. Default value is ONE.

S_REF_TS	Used to select between the NCO reference clock input, or the output side of the NCO as the clock which drives the timestamp counter. When set to 1, selects the NCO reference clock input as the source clock for the timestamp counter. The timestamp counter must have a reference clock input of 4.096MHz. Default value is ZERO.
NCO_RESET	When set to one, this bit causes the NCO counter to be reset to x00000000. The NCO is not normally reset, even during a hard reset of the chip. The lack of a natural reset for the NCO is to insure that there is always a clock output at V_CLK_OUT. The use of the NCO_RESET bit should be restricted to test environments, since it is likely to cause a glitch on the V_CLK_OUT signal. Note that NCO_RESET MUST NOT BE TIED TO PIN RESET, since this would prevent V)CLK_OUT from running during a board reset.

3.4 TX Descriptor bits:

LTS Latch TimeStamp: causes a timestamp latch event on transmit frames when this bit is set to 1

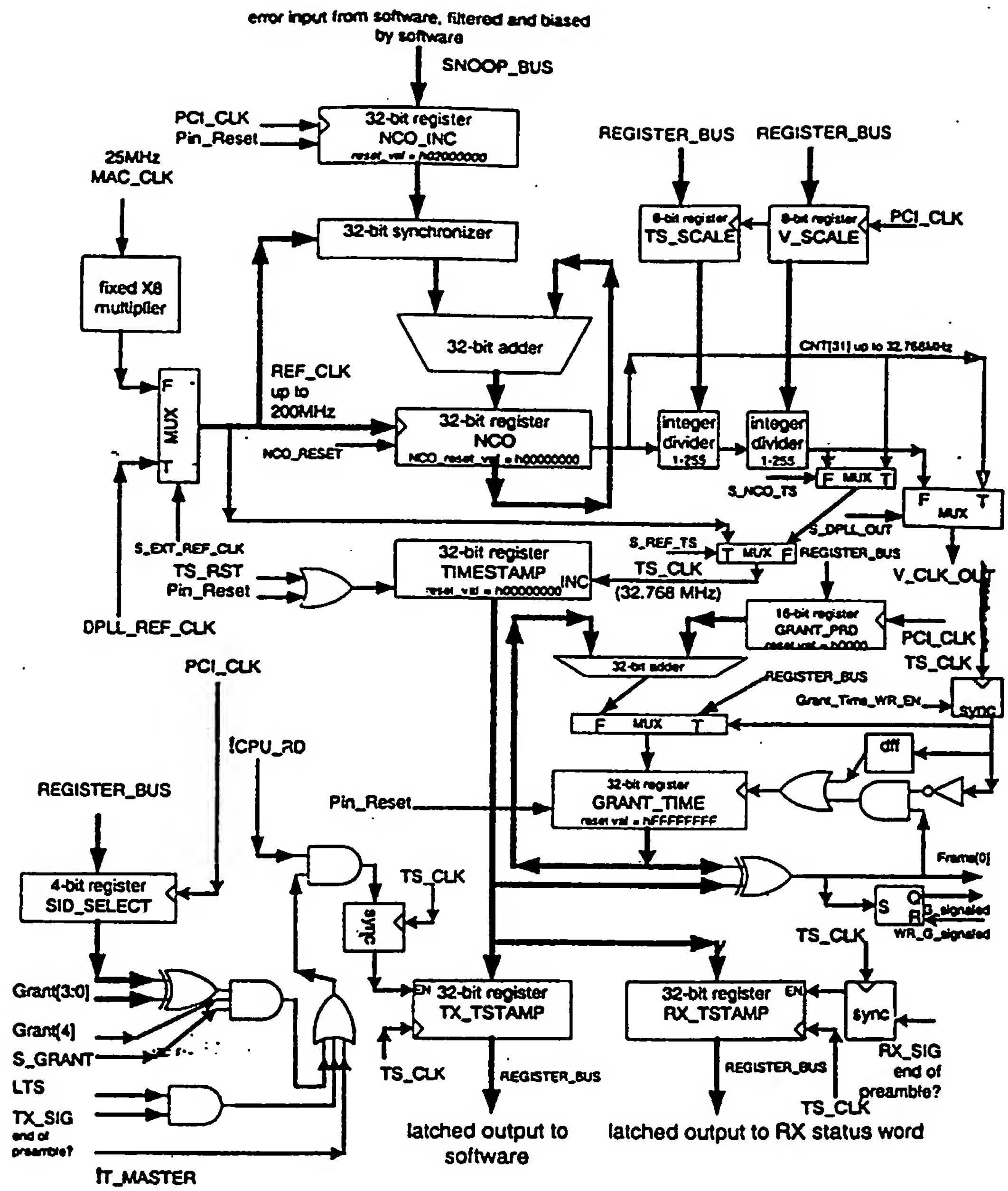
3.5 RX Descriptor bits:

RXTS[31:0] 32-bit receive timestamp value

3.6 HPNA TRC

The HPNA TRC circuit is based upon the TRC found in the BCM3350 and other devices. However, much of the circuit has been moved into software for the HPNA implementation. In fact, very little of the TRC remains in the HPNA version. In addition, new grant synchronization registers and logic are required by the HPNA MAC.

The following diagram describes the necessary components for the HPNA implementation:



The NCO error input is calculated by the device driver. The BIAS is added to the error, and the driver writes the resulting value to the NCO_INC register.

The correct BIAS value depends upon the V_CLK_OUT frequency requirement for the specific application.

The V_CLK_OUT signal must be square (50% duty cycle).

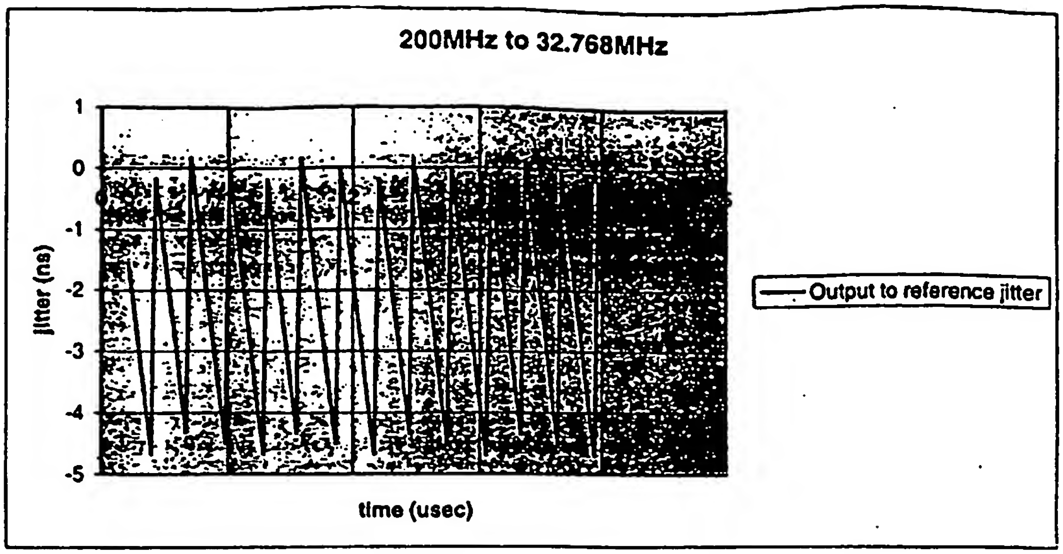
The V_CLK_OUT signal will begin with a default rate at power up. During RESET, the rate will be fixed. After RESET, the software will write values to various control bits that may change the rate of the V_CLK_OUT signal. These changes must not produce glitches on the V_CLK_OUT output.

The circuit as drawn allows V_CLK_OUT frequencies in the range: mear DC to 100MHz. However, because of the requirement for the timestamp to be running at 4.096MHz, an additional requirement must be placed on the V_CLK_OUT signal. The V_CLK_OUT signal must either be a ratio of integers divide of 4.096MHz, or it must be a ratio of integers multiple of 4.096MHz, where the integers must be in the range of 1-255, inclusive. This should provide sufficient range of V_CLK_OUT operation for all expected applications.

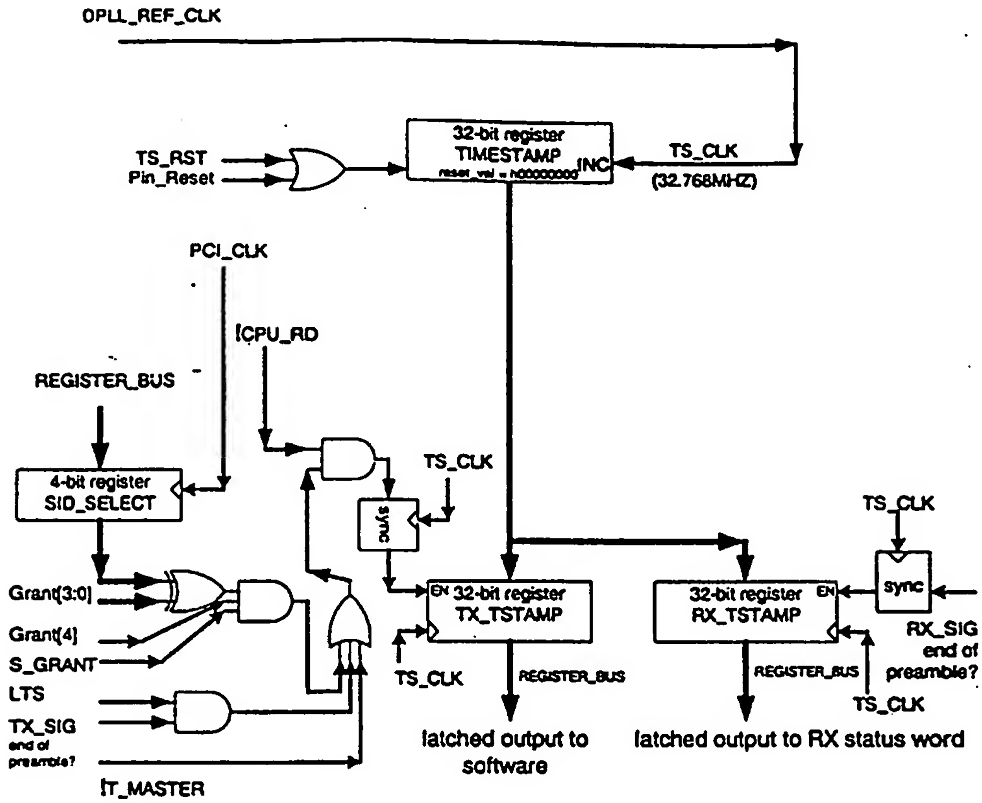
The accuracy of the DPLL decreases as the output frequency is reduced because the rounding error remains constant in magnitude, while the control word value decreases in magnitude. For a direct conversion of 200MHz to 8kHz, the control word for a 32-bit DPLL is 29F16, which produces a rounding error of 4ppm. If this rounding error is unacceptable, then any of several remediation steps can be taken, including, adding bits to the DPLL register. Adding 2 bits to the register changes the error to 1.1ppm. Another option is to perform less conversion in the DPLL, then feed the DPLL output to a divider to get the final output. It turns out that additional divide steps are required anyway, because a fixed rate clock is required for the timestamp function. The fixed rate for the timestamp is chosen to be 32.768MHz xxxx. (If the timestamps at the master and slave differ by a power of two, this would be acceptable, since software could accommodate the difference. Some other integer relationships are easy to adapt in a simple CPU – for example, the factor of 6 is easily obtained by two additions.)

The following chart shows the jitter in the DPLL output when the reference clock is 200MHz and the DPLL output clock (CNT[31]) is 32.768MHz. The jitter variance is +/- 2.5ns and the frequency of the jitter is about 3.3MHz. The jitter frequency is well above the audio range, and the +/- 2.5ns causes noise that is below –70dB in amplitude, thereby allowing the A/D to achieve the required 35dB SNR requirement of ITU-T recommendation G.712. Lower frequency components do exist in the jitter waveform, but the amplitude of these components is significantly lower than the 3.3MHz signal.

The offset of the jitter shown in the figure is corrected over time by DPLL frequency adjustments, such that the offset will ultimately vary around 0.


200MHz to 32.768MHz
1
0
-1
-2
-3
-4
-5
jitter (ns)
time (usec)
Output to reference jitter

3.6.1 Limited HPNA TRC implementation for 4220



This implementation will allow a timing master to be fully implemented. A timing slave will require an external DPLL and external grant signaling logic or a software approximation of grant signaling. (A software approximation of grant signaling would mean that software sets a timer to be interrupted when the next grant time arrives. The timer is set based on a read of the current timestamp as compared against the expected next grant time. The software would either initiate the framing and queuing process upon interrupt, or it would generate an output signal through a general purpose pin to cause external logic to create the frame. The accuracy of the grant timing on the slave device is not as critical as that required for maintaining a proper sample rate, since the queuing and contention delays are very highly variable anyway.)

The timing slave will have a single input, which is the DPLL_REF_CLK.

3.6.2 BCM4220 Pins

In the BCM4220 implementation, the timing slave output pins are deleted. In the BCM4220 timing slave configuration, the DPLL is external to the device.

PIN NAME	CM-side Function (HPNA timing master)		Handset Function (HPNA timing slave)	
DPLL_REF_CLK	Timestamp input clock	IN	Timestamp input clock	
Grant[4]	Grant Present Indication	IN	NA	
Grant[3]	Grant SID Value[3]	IN	NA	
Grant[2]	Grant SID Value[2]	IN	NA	
Grant[1]	Grant SID Value[1]	IN	NA	
Grant[0]	Grant SID Value[0]	IN	NA	

3.6.3 BCM4220 Registers:

3.6.3.1 TscControl (location 0x110)

Bit locations	Field name	Description
7-3	Reserved	
2	TsReset	When set to 1, forces timestamp register to value of 0x00000000. When set to 0, allows timestamp register to increment by one for each detected DPLL_REF_CLK rising edge.
1	SGrant	When set to 1, causes timestamp to be latched into txTimeStampHigh and txTimeStampLow registers whenever the value of tscSID matches the value of input pins Grant[3:0] and Grant[4] is asserted. When set to 0, disables txTimeStampHigh and txTimeStampLow latching under the stated conditions.
0	TMaster	When set to 1, enables txTimestampHigh and txTimestampLow registers to be latched with timestamp values at times determined by frame transmissions (through the LTS descriptor bit) or grant events (through the sGrant descriptor bit). When set to 0, enables txTimestampHigh and txTimestampLow registers to be latched with timestamp values at times determined by txTimeStampHigh and txTimeStampLow register read accesses.

Default value of this register is 0x05

3.6.3.2 TscSID (location 0x114) (=tscSID)

Bit locations	Field name	description
7-4	Reserved	
3-0	SID	SID value that is to be matched by Grant[3:0] pins in order to cause a grant timestamp value to be latched. When the Grant[3:0] pins match the SID value and the Grant[4] input is 1 and the sGrant register bit is 1, then the current timestamp value will be latched into the txTimeStampHigh and txTimeStampLow registers.

Default value of this register is 0x00

3.6.3.3 txTimeStampLow (location 0x118)

Bit locations	Field name	description
15-0	txTimeStampLow	Least significant 16 bits of the latched tx timestamp value

Default value of this register is undefined.

3.6.3.4 txTimeStampHigh (location 0x11a)

Bit locations	Field name	description
15-0	txTimeStampHigh	Most significant 16 bits of the latched tx timestamp value

Default value of this register is undefined.

3.6.3.5 rxTimeStampLow (location 0x11c)

Bit locations	Field name	description
15-0	rxTimeStampLow	Least significant 16 bits of the latched rx timestamp value

Default value of this register is undefined.

3.6.3.6 rxTimeStampHigh (location 0x11e)

Bit locations	Field name	description
15-0	rxTimeStampHigh	Most significant 16 bits of the latched rx timestamp value

Default value of this register is undefined.

3.6.4 New BCM4220 TX Descriptor bit:

Bit 25	LTS	Latch TimeStamp: causes a timestamp latch event on transmit frames when this bit is set to 1

3.6.5 New BCM4220 RX Descriptor bits:

Byte 27	rxTimeStamp[31:24]	MSbyte of rxTimeStamp
Byte 26	rxTimeStamp[23:16]	upper middle byte of rxTimeStamp
Byte 25	rxTimeStamp[15:8]	lower middle byte of rxTimeStamp
Byte 24	rxTimeStamp[7:0]	LSbyte of rxTimeStamp

4 Synchronization Software

The circuit that has been implemented in the 4220 device requires software control to complete the timing synchronization function. With the same circuit, HPNA network nodes will be able to operate as one of two types at any given time. Nodes will either function as a timing master, or as a timing slave. There may be more than one timing master active at any given time on a particular HomePNA LAN. Timing master and timing slave nodes have different physical connections and must be serviced by software in differing manners. The behavior of the software algorithm for each type of node is described in the following sections.

4.1 Timing Master Operation

The timing master will perform the following tasks:

1. Initialize the device as a timing master
2. generate pairs of TRM packets at 1 second intervals
3. generate a pair of TRM in response to a received TQM
4. generate a TRM in response to a the establishment of a new channel for a given MAC address, or in response to a received TSM (TRM in this case does not need to be a pair)
5. generate a TRM with the lost-lock indication when lock has been lost at the Cable Modem or other source of reference timing information (such as a DSL modem)

4.1.1 Initialization of the Timing master

Set the tMaster bit of the control register to force the device to operate as a timing master. Reset the sGrant bit of the control register. Initialize TRM sequence number space to x0000.

4.1.2 TRM pair generation

TRM pairs are sent using a period of at most one second. TRM pair generation is as follows:

Create a TRM message with TRM_type = x00 and with TRMSeqNum set to the next unused TRMSeqNum. Set PrevTRMSeqNum to x0000. Set Timestamp to x00000000. Set NumGrants to x00. Destination address is fixed as the broadcast address.

Queue the TRM in the TX queue of the 4220 with the LTS descriptor bit set to 1.

After the TRM is reported to have been transmitted, read the value latched in the TX_TIMESTAMP register. Create a new TRM with TRM_type = x00, TRMSeqNum set to the next unused value. PrevTRMSeqNum must be set to the value of TRMSeqNum in the first TRM of the pair. Timestamp should be written with the value of TX_TIMESTAMP that was just read from the BCM4220. NumGrants is set to x00.

DFPQ priority of all TRM is set to 6.

Queue the second TRM in the TX queue of the 4220 with the LTS descriptor bit set to 0.

4.1.3 master receives a TQM

The reception of a TQM is a request by a timing slave for the immediate transmission of a pair of TRM. The master must respond by immediately executing the TRM pair generation procedure. The normal 1 second periodic timer should not be disturbed.

4.1.4 Generate a TRM with grant timing information

A TRM may include Grant Timing information. Not all TRM are required to include grant timing information. A TRM with grant timing information must be generated in response to either of two events.

1) a latency-sensitive service flow is initialized (e.g. a VoIP connection is established)
2) a TSM is received

In either case, the TRM is constructed in the following manner:

First, Grant timing information is obtained:

The timing master keeps a list of MAC addresses and their associated SIDs. SIDs are Service Flow ID's that are assigned by the cable modem head end equipment when the VoIP connection is set up. The cable modem software must track all currently active SID values and keep a table which associates each value with an HPNA LAN MAC address. When a TSM is received, the timing master must get all channel ID's associated with that MAC address and then gather grant timing information for each channel ID.

Grant Timing information is obtained through the following mechanism:

The driver insures that no outstanding LTS bit remains set in the active TX descriptor list.
A selected channel ID (SID value) is placed into the tscSID register of the 4220.
The current value of the TX_TIMESTAMP register is read and stored.
The sGrant register bit is set.
The driver waits 10 msec (or whatever time is appropriate for the given channel – the wait time is equal to the period of the traffic flow).
The driver reads the TX_TIMESTAMP register and compares it to the stored value.
If the values differ, then the driver assumes that a valid timestamp has been captured for the selected SID.
If the values are the same, then the driver waits for the period of the flow and reads the TX_TIMESTAMP again.

The sGrant register bit is cleared.

The TRM is constructed as follows:

Create a TRM message with TRM_type = x00 and with TRMSeqNum set to the next unused TRMSeqNum. Set PrevTRMSeqNum to x0000. Set Timestamp to x00000000. Set NumGrants to x01. Destination address is set to the broadcast address.

MACAddr is set to the MAC address of the requesting node.
Channel_ID is set to the appropriate channel ID.
Gtimestamp is set to the value read from the TX_TIMESTAMP register.
The LTS bit of the TX descriptor is set to 0.

DFPQ priority of all TRM is set to 6.

The driver may choose to collect grant timing information for multiple channel_ID's for a given MACAddr before creating a TRM with grant timing information. However, it is best to deliver the grant timing information for any channel as quickly as possible.

Note that the tscSID register is loaded with a different value depending upon whether the device is attached to a BCM3308 or a BCM3350 cable modem device. BCM3308 SID values are positionally coded in the tscSID register. E.g. SID value of x3 corresponds to tscSID value of x8. For the BMC3350, SID values are directly represented in the tscSID register. E.g. SID value of x3 corresponds to tscSID value of x3.

4.1.5 When the Master loses lock

There needs to be an indication from the master reference clock source indicating a loss of lock. When this occurs, the master follows the same procedure as for sending TRM pairs, but with the TRM_type set to x01 instead of x00.

4.2 Timing slave operations

Timing slave devices will receive clock and grant timing information from timing master devices. Timing slaves will use this information for two purposes. The clock information will be used to keep the local clock locked to the master clock. The grant timing information will be used to determine when to frame a set of voice samples and send the frame to the CM.

There are several local variables to be maintained by the slave software. They include:

NCO_BIAS – the nominal divider for the NCO that translates the 200MHz reference crystal to the timestamp clock frequency (nominally 32.768MHz).
SLAVE_OFFSET – the difference between the master clock timestamp value and the slave timestamp value
Frequency_adjustment – the long-term estimate of the slave's frequency error from the master reference, smoothed with a filtering function
integrator_gain – coefficient for smoothing of the frequency_adjustment term
Phase_adjustment – the instantaneous adjustment to the slave's frequency error from the master reference, multiplied by the linear_gain term
linear_gain – coefficient for smoothing of the phase_adjustment term

The detailed relationships of these terms will be explained in later sections.

4.2.1 Initialization of the timing slave

Reset the tMaster bit of the control register to force the device to operate as a timing slave.
Set the NCO_BIAS to the value of

$$\text{NCO\_BIAS} = \frac{2^{32} * f_{TS}}{200}$$

Where f_{TS} is equal to the desired Timestamp frequency in Megahertz. f_{TS} is fixed at 32.768 for this application. With this value for f_{TS}, the NCO BIAS is x29F16B12.
Set the frequency_adjustment to ZERO.
Set the integrator_gain term to 0.02 (TBD xxxx)
Set the phase_adjustment to ZERO.
Set the linear_gain term to 0.90 (TBD xxxx)
Set the SLAVE_OFFSET to ZERO.

4.2.2 Initialization of frequency_adjustment

In order to allow for frequency synchronization, the timing slave device incorporates a DPLL. The DPLL reference input has a nominal frequency of 200MHz. The reference clock drives an NCO which yields a clock with a reduced frequency which is intended to track the master's clock. The initial BIAS value for the NCO was calculated based on the assumption that the reference clock is at exactly 200MHz and the master clock is running at exactly 32.768MHz.

However, the actual reference clock value is only *nominally* equal to 200MHz. The typical crystal supplying the slave reference time has an error of +/- 100ppm. This error offset must be measured, and the NCO_BIAS value must then be corrected for this error. The local reference frequency error can be measured directly by simply comparing the master's TRM interval measurement with the slave's. When any TRM pair arrives, the master will indicate the current time; With knowledge of the master time from a previously-received TRM pair, it is possible for the slave to determine the amount of time that has passed, assuming that the master's clock is correct. Then the slave can examine its own estimate of the time that has passed during that same interval to determine the local error. If M_x is the master timestamp at time T_x and S_x is the slave's timestamp value at time T_x, then the following equation describes this method:

$$\text{Slave\_Frequency\_Error} = \frac{S_2 - S_1}{M_2 - M_1} - 1$$

Since the error could be quite small, the slave will have to wait for a long enough period of time to accurately measure it. With the timestamp accuracy at 30.5ns (at each end, using 32.768MHz as the timestamp clock), each reported timestampcan be inaccurate between 0 and 0.06 usec. Assuming a required tracking error of less than 1 ppm, the slave would have to measure the master/slave time difference over an interval greater than 0.06μsec/1ppm = 0.06 seconds = 60 milliseconds in order to insure that the frequency error had been measured to greater than 1 part in 100. I.e. after 60 msec. the frequency drift error contribution would be 6 usec and the measurement error would be -0/+0.06 usec. It is convenient to wait much longer than this, so that the error contribution due to timestamp resolution is greatly reduced. If the slave waits the normal 1 second TRM interval, then the measurement error is very small compared to the maximum desired tracking error of 0.52ppm. (The measurement error falls to than 0.06ppm.)

In any case, the first step for the timing slave is to wait for the arrival of *two pairs* of TRM. When the first pair of TRM arrives, the timing slave stores the master and slave indicated timestamps and waits. (The first TRM of the pair yields a slave timestamp, the second of the pair reveals the master timestamp for the same event.) When the next pair of TRM arrives, the slave calculates the slave frequency error as described above. A division operation is necessary for the calculation, but the division only needs to be performed during initialization. The operation is not time-critical. The frequency error needs to be translated to an NCO BIAS adjustment value in order to allow the NCO to be adjusted to the proper frequency. The result is the initial value for the frequency_adjustment variable:

Frequency_adjustment = NCO_BIAS * Slave_frequency_error

The integrated_gain term is not applied during the initialization step.

The frequency_adjustment will be added to the NCO_BIAS term and the phase_adjusment term to create the NCO control word.

An additional error exists because the master timing reference has some non-zero meandering component which is due to the cable modem's attempts to maintain frequency lock to the head end timetstamps. Once the cable modem's clock is locked, this meandering should not exceed about 1ppm. The error is small enough to ignore during the initialization step – after initialization, we can assume that the slave and master are *closely* locked. The remaining error will disappear in a short time during the tracking phase.

4.2.3 Timestamp Acquisition

Timestamp acquisition is the process whereby the timing slave determines the relative offset between the local time and the master time. Timestamp acquisition at the timing slave node is performed as follows:

Once the frequency_adjustment has been initialized, the master and slave timestamp clocks are declared to be in sync. Therefore, the indicated master and slave timestamps for the second received pair of TRMs that was used to calculate the initial frequency_adjustment value give the nominal clock offset. This offset must be stored in the SLAVE_OFFSET variable and is used by the slave to calculate any needed reference times.

SLAVE_OFFSET = $S_2 - M_2$

The SLAVE_OFFSET value is not used to modify the DPLL, nor is it used to modify the slave's timestamp register. SLAVE_OFFSET will never be updated, because the DPLL will attempt to track the master timestamp and keep the offset constant. Any master time that must be signaled to the VoIP circuit (such as a grant indication to determine framing) will be converted to an equivalent slave time first by adding the SLAVE_OFFSET value, and then the slave time will be signaled to the VoIP circuit.

Note that under normal circumstances, the timing slave will return a timestamp for every RX frame. The timing slave should preserve the timestamp which corresponds to the most recently received TRM frame in order to be able to calculate interval durations as needed.

4.2.4 Initialization of phase_adjustment

The initial phase_adjustment that would be calculated from the second pair of TRM would be zero, because the master and slave are declared to be locked in phase at that point in time (i.e. at initial sync time). As a result, there is no phase_adjustment necessary until the third pair of TRM is received – and even then, only if a measurable error has accumulated. So the initial value of the phase_adjustment term remains ZERO.

4.2.5 Initialization of NCO control word

The initial NCO control word is calculated with the initial frequency_adjustment and phase_adjustment terms along with the NCO_BIAS value:

NCO_Control = NCO_BIAS + frequency_adjustment + phase_adjustment

The NCO_control word is written to the NCO control register at the completion of the initialization step. In the BCM4220, the NCO is not implemented. The NCO control register is external to the device.

4.2.6 Tracking

The tracking function measures the error from the most recent TRM interval and then attempts to correct for that error in the next TRM interval. The error is corrected by modifying the frequency and phase adjustment terms based on the current error and then updating the NCO control word.

Following the arrival of any TRM pair, the current slave timestamp error is determined:

Curr_slave_error = $S_x - M_x$ – SLAVE_OFFSET

Where S_x is the slave timestamp for the current TRM pair and M_x is the master timestamp for the current TRM pair.

For each TRM interval, the interval duration is determined:

Curr_interval = $M_x - M_{x-1}$

The phase adjustment for a given interval is calculated as follows:

Phase_adjustment = linear_gain * NCO_BIAS * curr_slave_error/curr_interval

The frequency adjustment for an interval is calculated as follows:

Frequency_adjustment = frequency_adjustment + int_gain * NCO_BIAS * curr_slave_error/curr_interval

Where int_gain = integrator_gain.

One could continue to use the equation:

$$\text{Slave\_Frequency\_Error} = \frac{S_x - S_{x-1}}{M_x - M_{x-1}} - 1$$

to determine the frequency error for a given interval and then substitute this value for the curr_slave_error/curr_interval term in the given frequency_adjustment equation. But the curr_slave_error/curr_interval term gives an adequate approximation, even with aggressive values for the integrator_gain term. The assumption is that the slave remains fairly well-locked to the master, and in that case, the approximation holds. By using only one equation, an extra divide operation is avoided.

After modifying the adjustment values, the NCO control word is recomputed and reloaded into the DPLL:

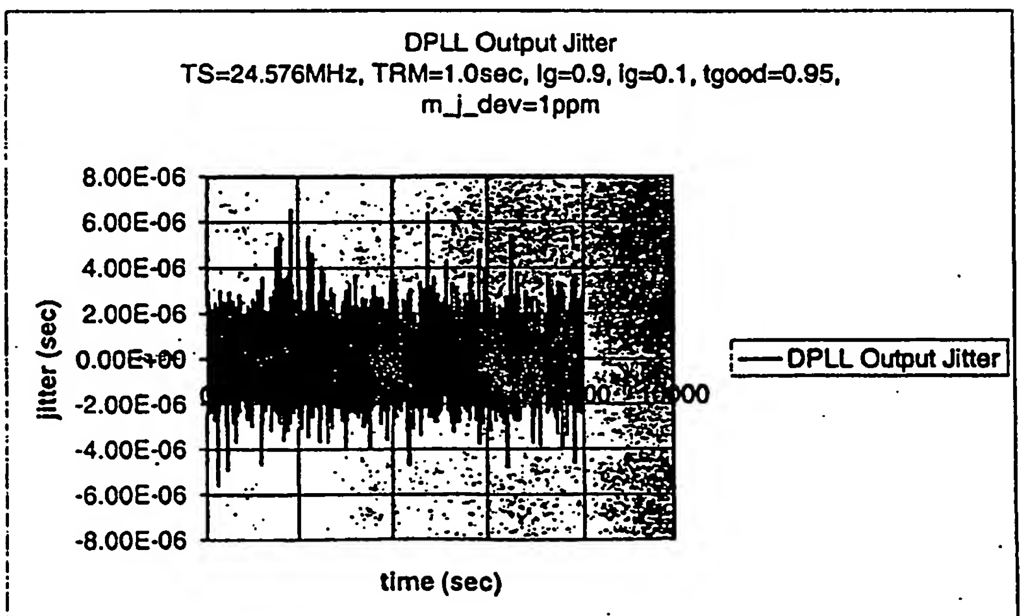
NCO_CONTROL = NCO_BIAS + frequency_adjustment + phase_adjustment

If the timing master creates TRM intervals of consistent 1 second times (with low jitter), then an additional math operation can be avoided by assuming that the curr_interval value is always equal to 1 second. Given that the TRM frames are sent with LL priority 7 (=DPFQ priority 6), the delivery latency jitter of a TRM should be well below 10 msec with 99% confidence. If a TRM pair is missing, then the original math operation needs to return, since the next interval will be an integer multiple of 1 second, requiring division by something other than 1. (As a further simplification, errors measured during longer intervals could be ignored, thereby avoiding this problem.)

There is the possibility of missing timestamp messages during normal tracking. The separation of crystal offset error from master-slave drift, NCO rounding error and reference source jitter is required in order to allow for free-wheeling NCO operation when no correction information exists for an interval. During intervals for which a TRM pair is lost, the NCO should be clocked at the nominal NCO BIAS plus the frequency error adjustment. (I.e. phase_adjustment should be reset to ZERO.) The frequency adjustment is unmodified in such circumstances. When a valid pair of TRM does arrive, the phase error that accumulated during the free-wheeling operation will be corrected in roughly a single TRM interval (depending upon the linear_gain term).

The chart shows the performance of the circuit with the following parameters:

The timestamp clock frequency is 24.576MHz.
The nominal TRM interval is 1.0 sec.
The linear gain is 0.9 over the nominal TRM interval.
The integrated gain is 0.1 over the nominal TRM interval.
The number of TRM pairs that arrive at the slave correctly is 95%.
The jitter in the master clock is +/- 1ppm corresponding to +/- 1 sigma, using normal distribution.
TRM interval jitter is corrected in making phase and frequency adjustments.

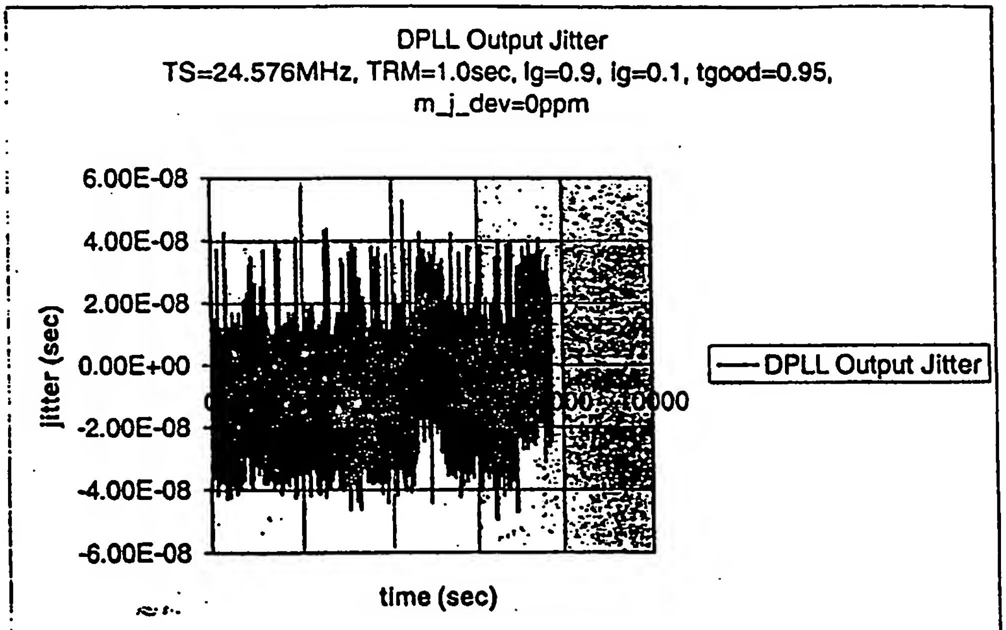


The simulation models a master clock jitter which is probably worse than will be encountered in reality, since the master clock will be created by a DPLL with correction intervals of 200 msec (MAX), while the simulation assumes master clock corrections which occur at 1 sec intervals. In the real system, the higher correction rate for the master clock will likely cause smoothing of the master clock jitter as observed by the slave. Also, it is expected that the CM clock will contain much less than 1ppm jitter over intervals of several seconds.

In general, the behavior of the circuit is very good, with the jitter shown fundamentally reflecting the jitter in the master clock input signal, with some amplification due to the timestamping inaccuracy and the fact that the slave system can only correct for past errors. It is impossible to construct a circuit which anticipates and corrects for future master clock jitter.

Note that in all cases, the behavior of the circuit modeled is to not offer a phase correction in the absence of any received TRM.

The second chart shows the tracking behavior of the DPLL when there is no master jitter, as a means of illustrating the performance of the DPLL in the presence of a stable master reference. ***Note the two orders of magnitude change in the vertical scale from the previous chart.***



4.2.7 Master loses lock

In the case when the cable modem completely loses lock, communication from the cable modem to the head end is disallowed. When this loss of synchronization occurs at the timing master, lost-lock TRMs will be sent to timing slaves so that they do not attempt to track the master clock. When the timing master re-acquires lock, the master must resume sending TRMs with a locked indication. Timing slave devices noting the transition from lost-lock to locked state must perform a new acquisition cycle. During the period of lost lock, the slave may choose to continue to send the VoIP frames, since the master may recover quickly enough to send some of them.

4.2.8 Reception of Grant Timestamps

The GRANT_SIGNALED bit is cleared to zero.
The timing slave adjusts the received grant timestamp value with the SLAVE_OFFSET value.
An integer multiple of the grant period is added to the result and the final value is written to the GRANT_TIME register.
The software sets timer for just over one grant period.
After the expiration of the timer, the software checks the GRANT_SIGNALED bit. If set, then the grant is being properly signaled to the framing logic. If not set, then the software must add additional integer multiples of grant period to the originally received grant timestamp value and repeat the previous steps.

In the BCM4220, the grant signaling logic is absent. In this case, the grant timing must be approximated by a software timer which is based on the estimated time to the next grant. The grant indication (framing) output would be signaled through a general purpose I/O pin.

5 Open questions:

Q: What about a two-line unit – how do you support multiple grant indications to each different call? Add new GRANT_TIME registers – use one incrementer to add 10msec to each, add new outputs for each new line.

Q: What clock output frequencies are needed at the handset?

For the AD73311, the input clock needed is 8.192MHz.
For the AM79C02, a 4.096MHz clock is needed.
For any linear codec, any of 8, 16 or 32 kHz would be needed.

The wide variation in required A/D conversion frequency reference creates a requirement for additional divide stages beyond the DPLL itself.

Q: What frequency range should be supported as reference clock inputs to the DPLL?

The timing master will be driven from the CM's smooth clock output which is intended to be a synchronous derivative of the DOCSIS clock for use by CM A/D equipment. The voice A/D equipment may have an input clock requirement of anything from baseband 8kHz to 32.768MHz. This is the expected frequency range for the clock which will be driving the timing master's timestamp function.

Q: Is it possible to have two timing masters in a home network – what would the slaves do in response?

Q: Can smaller integer dividers be used?

Q: There is a RESET problem that must be resolved. Either the NCO register is not reset at all, and there is a default clock provided during RESET that allows a DSP connected to V_CLK_OUT to be properly reset, or the NCO register is reset by an internal reset which is very short compared to the pin reset. The first case produces a desirable reset for the DSP, but doesn't allow easy testing, because the relative phase of the NCO is unknown. The second case requires some mechanism for producing a shortened internal reset – it might be possible to build a counter that counts out some number of reference clocks before releasing the internal shortened reset, but the question remains – what is a sufficiently large enough number of clocks, but at the same time, not too small that the clock is not yet stable?

One possible additional solution to the problem is to have the RESET of the NCO be attached to a register bit somewhere. Then, when the device is being tested, the NCO can be reset at will. This will cause an instantaneous phase jump in the V_CLK_OUT signal. If this same mechanism is desired in the lab, then there could be an additional circuit for smoothing the V_CLK_OUT during this reset operation to avoid a glitchy clock signal to the DSP.

Q: should TRM messages be bridged? What about HPNA-HPNA bridge connecting one HPNA LAN segment to another, and only one of the segments is connected to the WAN?

Q: what should be done in the case where two timing masters exist on one LAN segment?

Q:

6 Locations of interesting files

Files for Cable Modem TRC implementations:

\\Fs-irva-02\Projects-V2\BCM3000\BCM3350\work\sjkim\rtl\usphy
VoIP documents web page:

http://home/doc/VoIP/index.html

7 Timestamp Report Message (TRM)

The Timestamp Report Message protocol is intended to convey system-level timing information between two nodes of a HomePNA network. One node is assumed to be the timing master, and the other node is a timing slave. There may be more than one timing slave for a given timing master.

Timing master devices send timestamp messages to timing slaves on a periodic basis. Timing slaves use the timestamps to synchronize a local clock to the timing master's clock.

The TRM protocol also supports the conveyance of specific time information relating to connection-based service flows. In particular, the desired arrival time of a packet transferred from timing slave to timing master may be conveyed ~~from a timing master~~ to a timing slave device through the TRM protocol.

The TIMESTAMP REPORT message (TRM) is a ~~newly defined~~ Link Control Frame of SStype=TBD6, as follows:

Field	Length	Meaning
DA	6 octets	Destination Address (FF.FF.FF.FF.FF.FF)
SA	6 octets	Source Address
Ethertype	2 octet	0x886c (HPNA Link Control Frame)
SSType	1 octet	=TBD6
SSLength	1 octet	Number of additional octets in the control header, starting with the SSVersion field and ending with the second (last) octet of the Next Ethertype field. Minimum is 16.
SSVersion	1 octet	=0
TRM_type	1 octet	Value of x00 means that this is a TRM containing a valid timestamp. Value of x01 means that the master does not have a valid clock and slaves should give local indication that they are no longer locked to a master reference. Value of x80 means that this is a TQM. Value of x81 means that this is a TSM. All other values are reserved.
TRMSeqNum	2 octets	Timestamp Report Message Sequence Number for this message. Sequence number of x0000 indicates an initial TRM, implying that Timestamp and PrevTRMSeqNum are both invalid.
PrevTRMSeqNum	2 octets	Sequence number of TRM to which the Timestamp in this message is applicable. The value of PrevTRMSeqNum is *not* necessarily equal to TRMSeqNum minus one. PrevTRMSeqNum is set to x0000 for the first TRM of a TRM pair.
Timestamp	4 octets	Timestamp of a previously transmitted Timestamp Report Message, corresponding to PrevTRMSeqNum. The LSBit of the Timestamp corresponds to a time of 0.030517578125μsec = one clock tick at 32.768 MHz. The Timestamp will rollover every 131 seconds = 2.2 minutes.
NumSlots	1 octet	Number of Slot Timestamps specified in the payload of this control message. NumSlots may be zero. Each Slot Timestamp is accompanied by a MACAddr, and Channel_ID field. Including the Slot Timestamp, each Slot Timestamp is 12 bytes long.
PAD_0	3 octets	Padding to align to a 32-bit boundary. Always present, even when NumSlots has the value of 0.
MACAddr	6 octets	MAC Address associated with the immediately following Channel_ID and ~~GTimestamp~~STimestamp.
Channel_ID	2 octets	Identifier for a channel associated with the immediately preceding MACAddr.
STimestamp	4 octets	Slot Timestamp corresponding to the immediately preceding Channel_ID. This is the time at which the TRM sender wishes to receive a future constant bit rate service flow packet in order to minimize overall latency of delivery to a synchronous network. The time value corresponds to the time at the timing master. Additional packets for the identified service flow are expected to arrive at periodic intervals measured from this time. The LSBit of the STimestamp corresponds to a time of 0.030517578125μsec = one clock tick at 32.768 MHz.
MACAddr	6 octets	MAC Address associated with the immediately following Channel_ID and STimestamp.
Channel_ID	2 octets	Identifier for a channel associated with the immediately preceding MACAddr.
STimestamp	4 octets	SlotTimestamp corresponding to the immediately preceding Channel_ID. This is the time at which the TRM sender wishes to receive a future constant bit rate service flow packet in order to minimize overall latency of delivery to a synchronous network. The time value corresponds to the time at the timing master. Additional packets for the identified service flow are expected to arrive at periodic intervals

		measured from this time. The LSBit of the STimestamp corresponds to a time of 0.030517578125μsec = one clock tick at 32.768 MHz.
• • •		[additional instances of MACAddr, Channel_ID and Gtimestamp fields, until the number of Gtimestamp fields equals NumGrants]
Next Ethertype	2 octets	=0
Pad	max(0,44-SSLength) octets	Any value octet
FCS	4 octets	

Figure 74.1 Timestamp Report Message

~~Timestamp report messages shall not be repeated, nor are any of the timestamp values contained within them. That is, if there are several STimestamp values in one Timestamp Report Message, then none of these STimestamp values shall be repeated in subsequent Timestamp Report Messages. (Excepting the case of timer wrap.) xxxx Why can't I repeat STIMESTAMP values?~~

A pair of timestamp report messages (TRM) is sent every 1 second to allow for timing recovery.

When the first message of each pair is sent, a timestamp is recorded as the message is being transmitted onto the medium by the timing master ~~(by setting the LTS bit of the TX descriptor to 1)~~. The exact moment at which the timestamp for the TRM is sampled is not important – however, the consistency of the sample time is important. All TRM timestamps must be taken at a fixed time (master_timestamp_offset) relative to the time at which the first preamble symbol is transmitted onto the wire. The variation in the value of master_timestamp_offset can be no more than +/- 2 μsec. The absolute value of master_timestamp_offset must be greater than or equal to ZERO μsec and less than or equal to 64 μsec.

The timestamp that was recorded during the transmission of the first TRM of a pair is placed into the body of the second TRM. The second TRM is transmitted as soon as is possible following the first transmission. The second TRM of the pair does not require a timestamp to be recorded.

The number of Slot Timestamps in a TRM may be zero.

It is assumed that Slot Timestamp periods for each channel have been communicated through an out of band mechanism.

All timestamp protocol messages are sent with link layer priority of 7, which corresponds to DFPQ priority of 6 for all possible mappings.

Timing slave devices noting a transition of master state from lost-lock to locked must initiate an acquisition cycle when the transition is noted.

How to choose a master when there are several masters? Does the handset somehow get a master MAC address delivered to it once call agent discovery is performed at a higher layer? I.e. each handset will discover two gateways and choose one. This occurs at a higher layer than the TRM protocol. How does the TRM layer get the selected master's MAC address?

8 Timestamp Request Message (TQM)

The TIMESTAMP Request message (TQM) is:

Field	Length	Meaning
DA	6 octets	Destination Address (FF.FF.FF.FF.FF.FF)
SA	6 octets	Source Address
Ethertype	2 octet	0x886c (HPNA Link Control Frame)
SSType	1 octet	=6
SSLength	1 octet	Number of additional octets in the control header, starting with the SSVersion field and ending with the second (last) octet of the Next Ethertype field. Minimum is 4.
SSVersion	1 octet	=0
TRM_type	1 octet	Value of x80 means that this is a TQM.
Next Ethertype	2 octets	=0
Pad	MIN(0,40 - SSLength) octets	Any value octet
FCS	4 octets	

Figure 86.1 Timestamp Request Message

A timestamp request message is sent by a timing slave to request the delivery of a pair of TRM.

TQM messages are always sent to the broadcast DA. since only one timing master should be active on any HPNA LAN segment. Xxxx I'd feel much safer if it is directed. This would allow for extension to a multi-master case.

9 Timestamp Slot Request Message (TSM)

The TIMESTAMP Slot Request message (TSM) is:

Field	Length	Meaning
DA	6 octets	Destination Address (FF.FF.FF.FF.FF.FF)
SA	6 octets	Source Address
Ethertype	2 octet	0x886c (HPNA Link Control Frame)
SSType	1 octet	=6
SSLength	1 octet	Number of additional octets in the control header, starting with the SSVersion field and ending with the second (last) octet of the Next Ethertype field. Minimum is 4.
SSVersion	1 octet	=0
TRM_type	1 octet	Value of x81 means that this is a TSM.
Next Ethertype	2 octets	=0
Pad	MIN(0,40 - SSLength) octets	Any value octet
FCS	4 octets	

Figure 96.1 Timestamp Slot Request Message

A timestamp slot request message is sent by a timing slave to request the delivery of a set of TRM which contains a slot timestamp for each of the active channels associated with the requestor's MACAddr. The set

of TRM that is sent by the timing master in response to the receipt of a TSM may consist of a single TRM, or it may consist of more than one TRM.

TSM messages are always sent to the broadcast DA, since only one timing master should be active on any HPNA LAN segment. Xxxx I'd feel much safer if it is directed. This would allow for extension to a multi-master case.

37367/CAG/B600

3. HomePNA Adapter

The HomePNA adapter provides a means connecting additional POTS phones to the in home wire pair without effecting the operation of the POTS phone connected directly to the in house wire pair. This achieved by inserting a HomePNA adapter between each POTS phone and the HomePNA network from the cable modem.

The HomePNA has a similar architecture to the voice engine is shown in FIG. 6. As shown in FIG. 39, the voice engine includes an HPNA analog front end (AFE) 1000 for connection to the existing wire pairs in the home. The HPNA AFE 1000 provides modulation of voice packets from an external telephony device 1002 to the in home wire pairs. The HPNA AFE 1000 also provides demodulation of voice packets from the in home wire pairs for further processing before delivery to the external telephony device 1002. The HPNA AFE 1000 can be implemented in a variety of technologies including, by way of example, an integrated circuit. An exemplary integrated circuit for the HPNA AFE 1000 is described in Section 2.1 herein.

The HPNA AFE 1000 is coupled to the HPNA MAC 1004. The HPNA MAC 1004 provides the framing and link control protocol for the voice packets exchanged between the external telephony device 1002 and the in home wire pairs. The HPNA MAC 1004 can be implemented in a variety of technologies including, by way of example, an integrated circuit. An exemplary integrated circuit for the HPNA MAC 1004 is described in Section 2.2 herein.

The HPNA MAC 1004 interfaces with a voice processor 1006 over a data bus 1007. The voice processor 1006 can be a ZSP DSP core with embedded communications software or any other technology known in the art. The described embodiment of the voice processor 1006 supports the exchange of voice, as well as fax and modem, between the single in home wire pair and the external telephony device 1002. The voice processor may be implemented with a variety of technologies including, by way of example, embedded communications software. A packet synchronizer 1012 synchronizes the processing of voice packets in the voice processor 1006 under control of the HPNA MAC 1004.

The embedded communications software enables transmission of voice, fax and data packets over packet based networks. The embedded software includes a voice exchange between a telephony device and the in home wire pair. The voice exchange provides numerous functions including, by way of example, echo cancellation to remove far end echos, DTMF detection, voice compression/decompression algorithms, jitter buffering to compensate for network jitter, lost frame recovery, and comfort noise generation during silent periods.

37367/CAG/B600

The embedded software may also include a fax image data relay between a standard Group 3 fax session and the in home wire pair. The fax relay provides increased bandwidth performance over traditional voiceband fax transmissions by invoking demodulation/modulation algorithms. The fax relay may also includes spoofing techniques during rate negotiation to avoid timeout constraints.

The embedded software may also include a modem data relay between an analog line connection and the in home wire pair. The modem relay provides increased bandwidth performance over traditional voiceband modem transmissions by invoking demodulation/modulation algorithms. The modem relay may also includes spoofing techniques during rate negotiation to avoid timeout constraints. The details of the described exemplary embodiment of the embedded software are discussed in Section 2.3 herein.

The SLIC 1010 interfaces with the CODEC to provide bi-directional communication between the external telephony device 1002 and the voice processor 1006. The CODEC 1008 includes an analog-to-digital converter (ADC) for digitizing voice from the external telephony device 1002 and a digital-to-analog converter (DAC) for reconstructing voice prior to delivery to the external telephony device 1002. The CODEC includes a bandlimiting filter for the ADC and a reconstruction smoothing filter for the output of the DAC. A sample synchronizer 1014 synchronizes the sampling rates of the DAC and ADC under control of the HPNA MAC 1004. Exemplary embodiments of the sample synchronizer 1014 and the packet synchronizer are described in more detail in Section 2.4 herein.

3.1 SLIC and CODEC

FIG. 40 is a block diagram of an interface between a SLIC assembly 100 and a CODEC 102 in accordance with an embodiment of the present invention. The SLIC assembly 100 communicates with the CODEC 102 over a transmitting (Vtx) interface 106 and a receiving (Vrx) interface 104 for transmitting and receiving, respectively, telephony data to and from the CODEC 102. Other data, such as SLIC control and ringing data, are communicated over a data interface 108. The SLIC assembly 100 typically interfaces with a telephony device for a full duplex bi-directional communication over tip and ring interfaces 110 and 112. The telephony device may include traditional analog telephones as well as digital equipment. For example, the digital equipment may be coupled to the tip and ring interfaces 110 and 112 through a modem (modulator-demodulator).

In one embodiment of the present invention, multiple SLIC assemblies may be fabricated on a single integrated circuit chip and/or packaged into a single integrated package. FIG. 41 is

a block diagram of a multiple SLIC assembly 150, which includes four SLIC assemblies integrated into a single package. As those skilled in the art will appreciate, a multiple SLIC assembly may include more or less number of SLIC assemblies than four. A CODEC 102 may include a single CODEC that interfaces with all four SLIC assemblies. Alternatively, the CODEC 102 may also include four individual CODEC's.

The multiple SLIC assembly 150 includes SLIC assemblies 100, 152, 154 and 156. A SLIC assembly 100 communicates with the CODEC 102 over transmitting and receiving interfaces 106 and 104. A second SLIC assembly 152 communicates with the CODEC 102 over transmitting and receiving interfaces 160 and 158. A third SLIC assembly 154 communicates with the CODEC 102 over transmitting and receiving interfaces 164 and 162. A fourth SLIC assembly 156 communicates with the CODEC 102 over transmitting and receiving interfaces 168 and 166. Each SLIC assembly 100, 152, 154 and 156 communicates with a telephony device assembly over tip and ring interface pairs, 110 and 112, 170 and 172, 174 and 176, and 178 and 180, respectively.

I. Advanced Differential SLIC Interface for Low Voltage Operation

The described embodiment of the SLIC assembly provides a differential interface to the CODEC. This approach results in a good signal-to-noise ratio and facilitates a high system level integration of CODEC functions with other system level resources that may otherwise be discrete.

FIG. 42 is a block diagram of a SLIC assembly and CODEC. The SLIC assembly 100 includes a SLIC interface circuit 200 between the CODEC 102 and a SLIC 202. The SLIC interface 200 provides an interface between the differential CODEC 102 and the single-ended SLIC 202.

The CODEC 102 interfaces with the SLIC interface circuit 200 over a differential interface. The differential interface includes a differential pair of receiving lines 204 and 206. Over these receiving lines, the SLIC interface circuit 200 receives telephony signals Vrx+ and Vrx-. The SLIC interface circuit 200 converts the received differential signals Vrx+ and Vrx- into a single-ended telephony signal Vrx, and provides it over a receiving line 224 to the SLIC 202.

The SLIC 202 provides a single-ended transmit signal Vtx to the SLIC interface circuit 200 over a transmitting line 226. The SLIC interface circuit 200 converts the received single-

37367/CAG/B600

ended transmit signal into a differential pair of transmit signals Vtx+ and Vtx-, and provides them to the CODEC 102 over a differential pair of transmitting lines 208 and 210.

The SLIC 202 also communicates directly with the CODEC 102. The CODEC 102 provides a battery select signal 212 to the SLIC 202. The battery select signal 212 is used to select between one of two selectable battery voltages for power savings. The SLIC 202 may also receive a power from the CODEC for its operation.

When a call is made from a remote resource to a telephony device during an on-hook condition, the CODEC 102 sends a ringing signal 222 to the SLIC 202. The SLIC 202 generates voltages for ringing on tip and ring interfaces 110, 112 providing an alternating current (AC) source to a telephony device. In response, the telephony device provides an indicator to a user, e.g., a bell on the telephony device, rings.

If the call is answered, e.g., by lifting a handset, while the telephony device rings, direct current (DC) loop detection is used to determine an off-hook condition when the handset is lifted. The DC loop is formed between the SLIC 202 and the telephony device over the tip and ring interfaces 110 and 112. When the SLIC 202 detects the off-hook condition, the SLIC provides a detect signal 214 to the CODEC. The CODEC, in response, stops sending the ringing signal 222.

The CODEC 102 in the described embodiment also sends data signals C0, C1 and C2 to the SLIC 202 over data interfaces 216, 218 and 220, respectively.

FIG. 43 is a circuit diagram of a SLIC interface circuit 200 in one embodiment of the present invention. The SLIC interface circuit 200 includes three operational amplifiers (op amps) 300, 324 and 342. The op amp 300 is used to convert differential receive signals Vrx+ and Vrx- received from a CODEC over receiving lines 204 and 206 into a single-ended receive signal Vrx. The op amp 300 provides the single-ended receive signal Vrx to a SLIC over a receiving line 224.

The op amps 324 and 342 are used to convert a single-ended transmit signal Vtx received from the SLIC over a transmitting line 226 into a differential pair of transmit signals Vtx+ and Vtx-. The differential transmit signals Vtx+ and Vtx- are provided to the CODEC.

In a receiving path, bias resistors 310 and 312 are coupled to the receiving lines 204 and 206, respectively. The other end of the bias resistor 310 is coupled to a positive voltage supply, e.g., Vdd, and provides biasing between the positive voltage supply and the receiving line 204.

The other end of the bias resistor 312 is coupled to a negative voltage supply, e.g., ground, and provides biasing between the negative voltage supply and the receiving line 206.

A shunt capacitor 314 is coupled between the receiving line 204 and the receiving line 206. A current-limiting resistor 316 is coupled between the receiving line 204 and an inverting input 304 of the op amp 300. A current-limiting resistor 318 is coupled between the receiving line 206 and a non-inverting input 302 of the op amp 300.

The non-inverting input 302 of the op amp 300 is also coupled to one end of a shunt capacitor 320 and one end of a bias resistor 322. The other ends of the shunt capacitor 320 and the bias resistor 322 are coupled to the negative voltage supply. Thus, the shunt capacitor 320 and the bias resistor 322 form a parallel RC-circuit between the non-inverting input 302 of the op amp 300 and the negative voltage supply.

The op amp 300 provides an output as the single-ended receive signal Vrx over the receiving line 224. The output of the op amp 300 is also fed back into the inverting input 304 through a capacitor 308 and a variable resistor 306 in parallel. The gain in the signal receiving path may be controlled by varying the resistance of the variable resistor 306.

The single-ended transmit signal Vtx received over the transmitting line 226 is provided to the op amps 324 and 342 for conversion into a differential pair of transmit signals Vtx+ and Vtx-, which are provided to the CODEC over the transmitting lines 208 and 210, respectively.

The single-ended transmit signal Vtx is provided over the transmitting line 226 to an inverting input 328 of the op amp 324 through a current-limiting resistor 336. A non-inverting input 326 of the op amp 324 is coupled to the negative voltage supply. An output of the op amp 324 is provided as the positive differential signal Vtx+ through a current-limiting resistor 334. The output of the op amp 324 is also fed back into the inverting input 328 through a capacitor 332 and a variable resistor 330. The gain of the op amp 324 may be adjusted by varying the resistance of the variable resistor 330.

The single-ended transmit signal Vtx is also provided to a non-inverting input 344 of the op amp 342 through a current-limiting resistor 354. The non-inverting input 344 is also coupled to the negative voltage supply through a shunt capacitor 350 and a variable resister 352, which form a parallel RC-circuit. The DC level of the differential transmit signals Vtx+ and Vtx- may be controlled by adjusting the resistance of the variable resistor 352. An output of the op amp 342

is provided as the negative differential transmit signal Vtx- over the transmitting line 210 through a current-limiting resistor 348. The output of the op amp 342 is also fed back into an inverting input 346 of the op amp 342.

A shunt capacitor 338 and a resistor 340 are coupled between the differential transmitting lines 208 and 210. Thus, the shunt capacitor 338 and the resistor 340 form a parallel RC-circuit between the differential transmitting lines 208 and 210.

II. DSP Based Switched Mode Class D SLIC

FIG. 44 is a block diagram of a DSP based SLIC in one embodiment of the present invention that uses class D switched mode amplifiers. The SLIC 202 receives a Vrx signal 224 and transmits a Vtx signal 226. The SLIC 202 also provides tip and ring interfaces 110 and 112. The SLIC 202 includes a DSP based modulator 400, a pair of Class D drivers 404 and 406, a pair of low pass filters 408 and 410, and a tip/ring sampling circuit 402.

In the described embodiment, in order to reduce power dissipation, the Class D drivers 404 and 406 are implemented under control of the DSP based modulator 400. Power reduction can be achieved by switching the current from the power source off and on rather than allowing continuous current flow. In other embodiments, other types of switched mode circuits may be used to switch the power source current off and on.

The DSP based modulator 400 measure the tip and ring voltages to synthesize desired AC and DC impedances for AC impedance matching, DC biasing and power control. The DSP based modulator 400 provides control signals 414 and 416, respectively, to the Class D driver 404 and the Class D Driver 406 to turn them on and off. The control signals 414 and 416 may include AC and DC impedance information for AC impedance matching, DC biasing as well as power control. With the Class D drivers either on or off, instead of operating continuously, power dissipation is typically reduced. In other words, the voltage across the Class D drivers is approximately zero or the current through the Class D drivers is approximately zero.

Outputs 418 and 420 of the Class D drivers 404 and 406 are provided to the low pass filter 408 and the low pass filter 410, respectively. With the switching action of the Class D drivers at very high frequencies relative to the desired output frequencies, the low pass filters 408 and 410 can attenuate undesirable high frequencies in the outputs 418 and 420, and provide low frequency signals to tip and ring interfaces 110 and 112, respectively.

The telephony signals provided to the SLIC 202 over the tip and ring interfaces 110 and 112 for upstream communication are received by the tip/ring sampling circuit 402. The tip/ring sampling circuit 402 processes the received telephony signals and provides a processed signal 412 to the DSP based modulator 400.

FIG. 45 is a circuit diagram of the described embodiment of the SLIC 202. The DSP modulator 400 provides the control signals 414 and 416 to the Class D drivers 404 and 406, respectively. The Class D drivers 404 and 406 have a similar structure in this embodiment. In other embodiments, however, the Class D drivers 404 and 406 may have different structures.

The Class D driver 404 includes a p-channel MOSFET (Metal Oxide Semiconductor Field Effect Transistor) 508 and an n-channel MOSFET 510. When used as switches, MOSFET's generally have an advantage over their bipolar counterparts in that turn-off time is not delayed by minority carrier storage since the current in field-effect transistors is typically due to the flow of majority carriers only. The MOSFET's 508 and 510 can be enhancement type and with VMOS (V-shaped MOSFET) design. The VMOS design may be used to fabricate both n-channel and p-channel MOSFET's. In other embodiments, the MOSFET's may be other types of MOSFET's such as PMOS or NMOS.

A gate of the p-channel MOSFET 508 is coupled to the control signal 414 from the DSP based modulator 400. A source of the p-channel MOSFET 508 is coupled to a positive voltage supply V+. The drain of the p-channel MOSFET 508 is coupled to a driver output 418 and a drain of the n-channel MOSFET 510. A gate of the n-channel MOSFET 510 is coupled to the control signal 414. A source of the n-channel MOSFET 510 is coupled to a negative voltage supply V-. The drain of the n-channel MOSFET 510 is coupled to the driver output 418 and the drain of the p-channel MOSFET 508.

The p-channel MOSFET 508 and the n-channel MOSFET 510 typically are not operating in a turned-on state at the same time. Based on the voltage level of the control signal 414, either the p-channel MOSFET 508 or the n-channel MOSFET 510 turns on.

When the voltage level of the control signal 414 is sufficiently low, i.e., V_{GS} (gate-to-source voltage) < V_{T1} (first threshold voltage), the p-channel transistor 508 turns on, providing a logic high voltage to the low pass filter 408 using the driver output 418. The low pass filter 408 provides a filtered output through a current-limiting resistor 524 as the tip signal output of the SLIC 202 over the tip interface 110. While the p-channel transistor 508 is operating in a turned-on state, the n-channel transistor 510 is typically at a turned-off state.

37367/CAG/B600

On the other hand, when the voltage level of the control signal 414 is sufficiently high, i.e., $V_{GS} > V_{T2}$ (second threshold voltage) the n-channel transistor 510 turns on, providing a logic low voltage to the low pass filter 408 using the driver output 418. While the n-channel transistor 510 is operating in a turned-on state, the p-channel transistor 508 typically is at a turned-off state.

The low pass filter 408 includes an inductive element 516 and a capacitive element 518. A first terminal of the inductive element 516 is coupled to the driver output 418. A second terminal of the inductive element 516 is coupled to a first terminal of the capacitive element 518, and is also provided as the filtered output. A second terminal of the capacitive element 518 is coupled to a negative voltage supply, e.g., ground.

The D Class driver 406 is structured similarly and operates similarly to the D Class driver 404. The D Class driver 406 includes a p-channel MOSFET 512 and an n-channel MOSFET 514. The DSP based modulator 400 provides the control signal 416 to the gates of the MOSFET's 512 and 514 to provide a driver output 420 to the low pass filter 410. The low pass filter 410 includes an inductive element 520 and a capacitive element 522. The low pass filter 410 is structured similarly and operates similarly to the low pass filter 408. The low pass filter 410 provides a filtered output through a current-limiting resistor 526 as the ring signal output of the SLIC 202 over the ring interface 112.

The tip/ring sampling circuit 402 includes a voltage sampling amplifier 528 and a current sampling amplifier 530. The tip signal is provided to a non-inverting input of the voltage sampling amplifier 528. The ring signal is provided to an inverting input of the voltage sampling amplifier 528. The voltage sampling amplifier 528 takes a difference between the tip signal 110 and the ring signal 112 and provides a voltage difference signal 504 to the DSP based modulator 400. The voltage difference signal 504 is received by an ADC 500 in the DSP based modulator 400. The ADC 500 converts the voltage difference signal 504 into digital format, and uses it to calculate the DC and AC impedances.

The filtered output of the low pass filter 408 is provided to a non-inverting input of the current sampling amplifier 530 through a current-limiting resistor 534. The filtered output of the low pass filter 410 is also provided to the non-inverting input of the current sampling amplifier 530 through a current-limiting resistor 540 The non-inverting input of the current sampling amplifier 530 is also coupled to the negative voltage supply, e.g., ground, through a bias resistor 538. The tip signal and the ring signal are also coupled to an inverting input of the current sampling amplifier 530 through a current-limiting resistor 536 and a current-limiting resistor 542, respectively.

A current difference signal 506 is provided by the current sampling amplifier 530 to the DSP based modulator 400. The current difference signal 506 is also fed back into the inverting input of the current sampling amplifier 530 through a feedback resistor 532. The current difference signal 506 is received by an ADC 502 in the DSP based modulator 400. The ADC 502 converts the current difference signal 506 into digital format and uses it to calculate AC and DC impedances of the SLIC together with the digitized voltage difference signal 504.

III. DSP Based SLIC Architecture With Current Sensing-Voltage Synthesis Impedance Matching and DC Feed Control.

FIG.49 is a block diagram of a DSP based SLIC assembly 600 coupled to a CODEC 602 in one embodiment of the present invention. The CODEC 602 can be a highly integrated device that performs all signal processing functions of the SLIC assembly 600. The CODEC 602 may be scaled down in size with emerging silicon or other process technologies for fabricating devices that have smaller dimensions. AC and DC impedance synthesis and control can be performed in the digital domain by the CODEC 602. The SLIC assembly 600 and the CODEC 602 may be used in VoIP applications.

The SLIC architecture illustrated in FIG. 46 has a DSP design with a high voltage SLIC assembly acting primarily as an analog buffer and all signal processing performed in the digital domain by the CODEC. The CODEC 602 may be implemented using scalable low voltage CMOS. The SLIC/CODEC combination provides the BORSCHT (battery feed, over voltage protection, ringing, supervision, coding, hybrid and test) functions.

The SLIC assembly and CODEC combination in the described embodiment can meet the overall system level analog transmission requirements of Bellcore TR-NWT-000057 and ETSI 300 standards as applicable to short loop applications. This embodiment can meet the requirements of Bellcore TA-NWT-000909 standard, while reducing power consumption. Measures can be taken to minimize power in the idle standby state as well as during off hook transmission and ringing, with the highest priority given to the power reduction in the idle standby state.

Bellcore TA-NWT-000909 specifically addresses short loop transmission and signaling requirements found in FITL (fiber in the loop) systems. Since no ubiquitous requirements exist for analog transmission and signaling for hybrid filter coax networks, TA-NWT-000909 forms a basis set of requirements for cable IP telephony.

37367/CAG/B600

In the described embodiment, the combination of the CODEC and the SLIC assembly include the following features. The CODEC and the SLIC assembly perform all battery feed, over-voltage protection, ringing, signaling, hybrid and test (BORSHT) functions. This embodiment can also be configured to exceed LSSGR and ITU central office requirements. DC loop characteristics and loop supervision detection thresholds can be software programmable in the CODEC.

The features of the CODEC and the SLIC assembly combination can also include off-hook detection and 2-wire AC impedance. Off-hook and ring-trip detectors have programmable thresholds. The described embodiment of can also provide ringing with no external hardware. Other features may include integrated ring-trip filter and software enabled manual or automatic ring-trip mode. This embodiment preferably supports loop-start signaling. The 2-wire interface voltages and currents can be monitored for subscriber line diagnostics. The CODEC and the SLIC assembly also may have built-in-test (BIT) modes.

The integrated line-test and self-test features of the described embodiment include: leakage, capacitance and noise test; loop resistance (A to B and to ground and battery) test; echo gain and distortion test; idle channel noise test; and ringing test. The CODEC and the SLIC assembly can also support on-hook transmission and power/service denial mode.

The described embodiment can be configured to be compatible with inexpensive protection networks, and accommodates low tolerance fuse resistors while maintaining longitudinal balance. The line-feed characteristics can be independent of battery voltage. The described embodiment can provide linear power-feed with power management and automatic battery switching. Only a 5V power supply and battery supply are typically needed. Other features may include low idle-power per line, -40 degree C to 85 degree C industrial operation and small physical size.

The SLIC assembly 600 is a high voltage device that mainly acts as a buffer between the low voltage signal processing circuitry, i.e., CODEC, and the high voltage subscriber loop side for outgoing and incoming signals. With a DSP based AC impedance synthesis loop, numerous applications may be realized through software/firmware assembly programmability of the desired output impedance of the SLIC for both real and complex impedances.

With DSP control over the DC operating points of tip and ring, the DC voltage level may be used to control the loop current for the desired operating conditions in the off hook status,

37367/CAG/B600

ringing and fault conditions, i.e., current limiting. For on/off hook states, tip and ring signals may be fed with different DC offsets to provided DC loop current and the necessary amplifier headroom to be able to drive the AC voice signal in a non-distorted manner.

With DSP control of the DC feed in the ringing state, the SLIC assembly may operate in a balanced or non-balanced ringing mode. In the balanced ringing mode, tip and ring signals can be driven with the same DC voltages but differential AC voltages, thus producing a balanced differential ringing signal. In the unbalanced ringing mode, the tip lead can be at zero Volt while the ring lead provides a negative DC bias and a high amplitude AC ringing signal, single-ended instead of differential.

Referring back to FIG.49, the CODEC 602 transmits a Vtx signal 606 and receives a Vrx signal 604 to and from a central office and interfaces with the SLIC assembly 600 to form a subscriber interface loop. The SLIC assembly 600 communicates with a telephony device through tip and ring interfaces 622 and 624.

The CODEC 602 sums the received Vrx signal 604 together with a DC voltage and a synthesized impedance, and provides the summed signal to the SLIC assembly 600 as a Vdac signal 608. The CODEC also provides a voltage reference 614 to the SLIC assembly. In addition, the CODEC 602 provides a control signal 620 to the SLIC assembly 600 to control operations of the SLIC assembly. The SLIC assembly also receives battery power signal Vbat 612.

The SLIC assembly provides a feedback Vm 610, e.g., metallic loop voltage, back to the CODEC 602. The CODEC monitors loop conditions using the metallic feedback signal Vm 610. Upon detecting an off-hook condition, the SLIC assembly provides a detect signal 616 to the CODEC. The SLIC assembly also sends a Vadc signal 618 to the CODEC 602. The Vadc signal may be a difference between the tip signal and the ring signal.

The SLIC assembly in the described embodiment does not require a programming impedance circuit attached to the SLIC assembly/CODEC combination. The impedance synthesis can be performed entirely by the CODEC through DSP processing. The DSP based SLIC assembly/CODEC combination, having control over the DC feed, may provide balanced and non-balanced ringing without external hardware, such as relays, to ground the tip lead. With all signal processing performed in the digital domain, the size of the SLIC die may be reduced and thus a lower cost part may be realized. With a smaller SLIC die size, more SLIC/subscriber channels per die may be implemented to reduce the overall system cost of providing multiple channels in a single package, i.e., saves multiple package cost and increases reliability.

37367/CAG/B600

In the described embodiment, the SLIC assembly can be implemented in a quad assembly format where each package includes four SLIC assemblies. In other embodiments, the SLIC assembly may be implemented in other formats.

FIG. 47 is a detailed block diagram of the CODEC 602 and the SLIC assembly 600. The CODEC 602 includes a digital-to-analog converter (DAC) 700 and two analog-to-digital converters 710, 720. The CODEC 602 also include adders 701, 722, a B filter 712 and a Z filter 708.

The Vrx signal 604 received by the CODEC is added in the adder 701 to Vdc, which is a DC voltage provided to the CODEC. The Vrx signal 604 and the Vdc is also added in the adder 701 with an impedance voltage VZT, which is generated by a Z filter 708. The Z filter 708 provides the VZt through a filtering capacitor 704. The filtering capacitor 704 operates as a high pass filter between the Z filter 708 and the adder 701.

The Z filter is coupled to the feedback signal Vm 610 from the SLIC assembly 600, and uses the feedback signal Vm to determine the appropriate VZt for impedance matching. The feedback signal Vm is converted to a digital signal by the ADC 710 prior to being provided to the Z filter. The Z filter 708 is also coupled to ground through a switch 718. The switch 718 is used to disable feedback during ringing by coupling the output of the ADC 710 directly to ground.

The feedback signal Vm 610 is provided by a feedback amplifier 730 in the SLIC assembly 600. To provide the feedback signal Vm 610, the feedback amplifier 730 receives signals from a tip amplifier circuit 728 and a ring amplifier circuit 734, including tip and ring signals 622 and 624. The feedback amplifier 730 takes a difference between the ring signal and the tip signal and provides as the feedback signal Vm 610.

The CODEC 602 also provides a reference voltage Vref to an off hook detector 732 in the SLIC assembly. The off hook detector also receives the feedback signal Vm from the feedback amplifier 730 to detect an off hook condition. Upon detecting the off hook condition, the off hook detector 732 provides a detection signal 616 to the CODEC.

The CODEC 602 also includes a B filter. The B filter 712 filters the received signal Vrx and provides it to the adder 722 to be subtracted from the Vadc signal supplied by an upstream transmitter 736 of the SLIC 600 through a filtering capacitor 738. The filtering capacitor 738 operates as a high pass filter. The Vadc signal is converted into digital format by the ADC 720 and provided to the adder 722. The difference between the digitized Vadc signal and the B

37367/CAG/B600

filtered Vrx signal is provided as the Vtx signal 606. The upstream transmitter 736 is coupled to and receives inputs of the tip and ring signals 622 and 624.

The upstream transmitter 736 takes a difference between the tip signal 622 and the ring signal 624, and provides as a Vadc signal 618 to CODEC through a filtering capacitor 738. The filtering capacitor 738 blocks the DC component passing only the AC component. The Vadc signal is analog-to-digital converted by the ADC 720 prior to being provided to the adder 722.

The Vdac signal 608 from the DAC 700 of the CODEC 602 is provided to a tip driver 726 of the SLIC 600. The tip driver provides the signal to the tip amplifier 728, which in turn amplifies the provided signal and outputs it as a tip signal over the tip interface 622. The ring amplifier receives the tip signal and provides a ring signal over the ring interface 624. Functions of the tip driver 726, the tip amplifier 728 and the ring amplifier 734 will be described in more detail below in reference to FIG. 48.

FIG. 48 is a circuit diagram of one embodiment of the SLIC assembly 600 illustrated in FIG. 47. The SLIC assembly 600 receives the Vdac signal 608, the Vbat signal 612, the Vref signal 614, ground through a bias resistor 856, and the control signal 620, and provides the feedback signal Vm 610 and the detector signal 616 as well as the tip and ring signals over the tip and ring interfaces 622 and 624. When a typical telephony device is coupled between the tip and ring interfaces 622 and 624, a current between the tip and ring interfaces is typically represented by im and an impedance between them is typically represented by ZI. For upstream transmission, the SLIC assembly 600 receives the tip and ring signals from a telephony device and provides the Vadc signal 618 through the filtering capacitor 738 to the CODEC.

The operation of the SLIC assembly 600 may be more easily understood by analyzing the signals as separate AC and DC components. The Vdac signal 608 is provided to the tip driver 726. The Vdac signal 608 is a composite signal having both AC and DC components including the received voice signal Vrx, the DC operating point Vdc and the impedance synthesis signal VZt.

The tip driver 726 and the tip amplifier 728 provide a programmable gain for the low (on/off hook) and high (ringing) voltage operating states. The tip and ring DC voltages during the on/off hook states may be represented by a set of equations. Note that, with the feedback signal Vm 610, the loop current may be regulated by the CODEC by lowering the tip to ring voltage, i.e., raising the Vdac DC voltage.

37367/CAG/B600

The tip driver 726 includes an operational amplifier (op amp) 800. The Vdac signal is provided to an inverting input of the op amp 800 through a resistor 812. An output of the op amp 800 is fed back to the inverting input of the op amp 800 through a feedback resistor 814. The resistors 812 and 814 set the gain of the op amp 800. In the described embodiment, the op amp is a unity gain amplifier and the resistors 812 and 814 have identical resistance values. A non-inverting input of the op amp 800 is coupled to ground through a bias resistor 816, which has a resistance value of, e.g., R/2b or (R/b)||(R/b). Thus, the tip driver 726 is configured as an inverting amplifier since the op amp 800 inverts the input signal with a gain of -(R/b)/(R/b) = -1. Therefore, the tip driver 726 inverts the Vdac signal 608 and provides it to the tip amplifier 728.

The tip amplifier 728 includes an op amp 802, which is used to driver the tip signal. The output of the tip driver, i.e., the inverted Vdac signal, is provided to a non-inverting input of the op amp 802 through a current-limiting resistor 824. An inverting input of the op amp 802 is coupled to ground through a bias resistor 822. The resistance values for the bias resistor 822 and the current-limiting resistor 824 are identical in this embodiment. For example, the resistance values may be RGb for both of the resistors.

The inverting input of the op amp 802 is also coupled to a first terminal of a bias resistor 818. A second terminal of the bias resistor 818 is coupled to ground through a switch 819. For example, the bias resistor 818 may have a resistance value of RGa. Thus, when the switch 819 is open, the resistance between the inverting input of the op amp 802 and ground is the resistance of the bias resistor 822, which may be RGb. When the switch 819 is closed, however, the bias resistor 818 is in parallel with the bias resistor 822 between the inverting input of the op amp 802 and ground. In this case, the resistance value between the inverting input and ground is equal to the resistance value of the bias resistors 818 and 812 in parallel. For example, if the bias resistor 818 has a value of RGa and the bias resistor 822 has a value of RGb, the resistance value of the resistor equivalent to those two resistors in parallel is equal to RGa||RGb = ((RGa) x (RGb))/(RGa + RGb).

The inverting input of the op amp 802 is also coupled to an output of the op amp 802 through a current-limiting resistor 826 and a feedback resistor 820 in series. The resistors 826 and 820 may have values of, e.g., Rf and RG, respectively. Since the inverting input is coupled to ground, the op amp 802 is a non-inverting amplifier. The gain G of the tip amplifier 728, therefore, is (resistance value of the feedback resistor 820 + resistance value of the bias resistor 822) / (resistance value of the bias resistor 822). For example, if the resistance values of the resistors 820 and 822 are RG and RGb, respectively, the gain G is equal to (RG + RGb)/(RGb) = 1 + (RG/RGb). An output of the tip amplifier is provided as the tip signal output over the tip interface 622.

37367/CAG/B600

The ring amplifier 734 includes an op amp 806, which is used to drive the ring signal. The op amp 806 provides an output, which is provided through a current-limiting resistor 846 as the ring signal output over the ring interface 624. A non-inverting input of the op amp 806 is coupled to a Vbat signal 612 through a current-limiting resistor 842. The non-inverting input of the op amp 806 is also coupled to ground through a bias resistor 836. An inverting input of the op amp 806 receives the tip signal 622 through a current-limiting resistor 844. The ring signal is fed back to the inverting output through a feedback resistor 848.

For example, in this embodiment, the resistance values of the resistors 836, 842, 844 and 848 are identical at R. The resistance value of the resistor 846, e.g., is Rf.

The op amp 806 of the ring amplifier 734 is configured to receive inputs of the Vbat signal 612 and the tip signal, which may be expressed as Vtip. Since the Vbat signal is coupled to the non-inverting input of the op amp 806 and the Vtip signal is coupled to the inverting input of the op amp 806, the ring signal, which may be expressed as Vring, is equal to Vbat - Vtip.

Therefore, relationship between G, Vtip, Vring and Vtip-ring may be represented by the following equations.

Eq. 3.1.1) $G = (1 + RG/RGb)$;

Eq. 3.1.2) $Vtip = -1 \times Vdc \times G$;

Eq. 3.1.3) $Vring = Vbat - Vtip = Vbat + (Vdc \times G)$; and

Eq. 3.1.4) $Vtip\text{-}ring = 2Vtip - Vbat = -1 \times (Vbat + (2Vdc \times G))$.

For example, for on hook and off hook DC states, if Vdc = 1V, RG = 390K, RGb = 78K and Vbat = -24V, then gain G = 6, Vtip = -6V, Vring = -18V and Vtip-ring = 12V.

During the ringing state, the gain of the tip amplifier is increased to provide a large DC level on tip and ring interfaces 622 and 624, thus resulting in higher ringing amplitude. By way of example, when the CODEC operates with 3.3V supply, a gain of 40 is desirable. For such increase in gain, the switch 819 is closed. In this case, equations to represent Vtip and Vring are identical to the equations 3.1.1 through 3.1.4 except that the gain is increased.

Therefore, relationship between G, Vtip, Vring and Vtip-ring may be represented by the following equations.

37367/CAG/B600

Eq. 3.2.1) G = (1 + RG /(RGa||RGb));

Eq. 3.2.2) Vtip = -1 x Vdc x G;

Eq. 3.2.3) Vring = Vbat - Vtip = Vbat + (Vdc x G); and

Eq. 3.2.4) Vtip-ring = 2Vtip - Vbat = -1 x (Vbat + (2Vdc x G)).

For example, using the equations 3.2.1 through 3.2.4, when Vdc = 1V, RG = 390K, RGa||RGb = 10K and Vbat = -80V, gain G = 40, Vtip = -40V, Vring = -40V and Vtip-ring = 0V.

The feedback amplifier 730 includes an op amp 804, which is used to drive the feedback signal Vm. An inverting input of the op amp 804 receives the output of the op amp 806 through a current-limiting resistor 838. The inverting input of the op amp 804 also receives the tip signal through a current-limiting resistor 832. In addition, the inverting input of the op amp 804 receives a feedback output of the op amp 804 through a feedback resistor 834.

A non-inverting input of the op amp 804 is coupled to ground through a bias resistor 828. The non-inverting input of the op amp 804 is also coupled to the output of the op amp 802 through a current-limiting resistor 830. In addition, the non-inverting input of the op amp 804 is coupled to the ring signal through a current-limiting resistor 840.

The output of the op amp 804, which is the output of the feedback amplifier 730, is provided to the CODEC as the feedback signal Vm 610. The feedback signal Vm 610 is also provided to the off hook detector 732.

The tip driver 726 and the tip amplifier 728 provide programmable gain for the low (on/off hook) and high (ringing) operating states. Since the Vdac has one DC component Vdc and two AC components Vrx and VZt, the equations that represent AC states are as follows.

Eq. 3.3.1) G = (1 + RG/RGb);

Eq. 3.3.2) Vtip = -G x (Vrx + VZt);

Eq. 3.3.3) Vring = G x (Vrx + VZt); and

Eq. 3.3.4) Vtip-ring = -2G x (Vrx + VZt).

Referring back to FIG. 48, the feedback signal Vm is equal to 2 x im x Rf/b, where im is the metallic loop current, Rf is the fuse resistance and b is the attenuator ratio, which typically is between 1 and 10. Thus, the feedback signal Vm is a scaled voltage representation of the metallic loop current im. This scaled voltage is sampled by the ADC 710 in FIG. 47, and then processed digitally through the Z filter to adjust the output impedance of the SLIC. Therefore, for example, the feedback signal summed with the received voice signal is therefore VZt = Z x Vm = 2 x im x Rf/b.

When Vrx equals zero, the AC output impedance is thus Zo = Vtip-ring / im, which results in Eq. 3.3.5) Zo = -2G x VZt = -4G x Z x Rf/b. For example, when Vrx = 0, RG = 390K, RGb = 78K, G = 6, Rf = 25 ohms, Z = 10, b = 10, and |Zo| = 600 ohms.

The received four wire to two wire (4w-2w) gain is calculated with Zxt = 0, therefore, in this case, Eq. 3.3.6) Gain 4w-2w = Vtip-ring / Vrx = -2G. With Zo matching the load, i.e., 600 ohms, the 4w-2w gain is reduced by a factor of 2, which results in Eq. 3.3.7) Gain 4w-2w = -G where Zo = Zload.

During the ringing state, the CODEC preferably shuts down the feedback signal from the SLIC by, for example, opening the switch 718 in FIG. 47. This typically will effectively eliminate the impedance matching functions to provide the maximum amplitude for ringing a telephony device such as a telephone. With G=40 during the ringing state, the 4w-2w gain is thus Eq. 3.4.1) Gain 4w-2w = Vtip-ring / Vrx = -2G. For example, when Vrx = 0.5VAC, RG = 390K, RGb||RGa = 10K, G = 40 and Rf = 25 ohms, Gain 4w-2w = -80 (with Zo = 0 ohms) and |Vtip-ring| = 40VAC. Note with 1VDC and 0.5VAC, this provides a 1V DC signal with a 1.4VPP AC riding on it at the output of the DAC. This would allow a common mode range of 0.3V to 1.7V at the output of the DAC, which should be consistent with a 3V process.

The upstream transmitter 736 is coupled to the tip signal and the ring signal through filtering capacitors 866 and 868, respectively. The filtering capacitors 866 and 868 operate as high pass filters. The upstream transmitter 736 composites the AC components of the tip signal and the ring signal, and provides the composite upstream voice signal to the CODEC 602 in FIG. 47. The hybrid balance function preferably is provided in the digital domain through digital signal processing (DSP) by the CODEC.

The upstream transmitter 736 includes an op amp 810, which is used to drive a Vadc signal. A non-inverting input of the op amp 810 is coupled to ground through a bias resistor 858. The non-inverting input of the op amp 810 is also coupled to the tip signal through a current-limiting resistor 862 and the filtering capacitor 866 in series. An inverting input of the op amp

37367/CAG/B600

810 is coupled to the ring signal through a current limiting resistor 864 and the filtering capacitor 868 in series. The output of the op amp 810 is fed back to the inverting input of the op amp 810 through a feedback resistor 860. The output of the op amp 810 is provided to the CODEC through a filtering capacitor 738, which operates as a high pass filter, as the Vadc signal.

The SLIC assembly 600 provides a low power loop monitoring function to alert the CODEC. The detector signal 616 is provided to the CODEC by the off hook detector 732. The off hook detector 732 includes an op amp 808. An inverting input of the op amp 808 is coupled to the feedback signal Vm through a resistor 854.

A non-inverting input of the op amp 808, which is used to drive a detector signal 616, is coupled to a reference voltage Vref through a bias resistor 850. The non-inverting input of the op amp 808 is also coupled to ground through a bias resistor 856. The bias resistor 856 can be a threshold resistor with the resistance value of, e.g., Rth. The non-inverting input of the op amp 808 is also coupled to the detector signal output 616 through a feedback resistor 852.

A logic low detector signal 616 is provided when loop current is received by the inverting input of the op amp 808 through the resistor 854, indicating an off hook condition. The detect threshold is set by the resistance value Rth of the threshold resistor 856, with hysteresis provided by the SLIC assembly.

Once the logic in the CODEC has been activated, the CODEC monitors loop conditions using the metallic feedback signal Vm 610, and provides filtering during the loop monitoring function. Note that it has been assumed that the CODEC, i.e., the DSP process in the CODEC, will monitor the loop current and provide the ring trip filtering and detection function since during the ringing state, the logic will be awake and active. A dial pulse function may also be monitored using the detection signal and/or through the DSP and the metallic feedback signal Vm.

FIG. 49A is a voltage graph 900 that illustrates a Vtip signal 908 and a Vring signal 904 during a balanced ringing mode in one embodiment of the present invention. In this embodiment, the signals have a negative DC bias 906 about which they oscillate during balanced ringing. FIG. 49B is a voltage graph 902 that illustrates a Vring signal 910 during a non-balanced ringing mode in an alternate embodiment. The Vring signal 910 oscillate about a negative DC bias 912. A Vtip signal remains grounded at 0V.

IV. Composite MOSFET Bipolar Complimentary Symmetry Driver with Local Feedback for Bias Stabilization

37367/CAG/B600

FIG. 50 is a block diagram of an op amp 1000. The op amp 1000 may be used as one or more of the op amps used in the SLIC assembly 600 of FIG. 48. The op amp 1000 may also be used in the SLIC interface circuit of FIG. 43, the SLIC assembly of FIG. 45 or any other circuit that uses op amps. The op amp 1000 receives inverting and non-inverting input signals Vin- 1010 and Vin+ 1012. The input signals are received by an input stage 1002 and provided to an output stage 1004. The output stage 1004 preferably includes a driver stage 1006 for driving an output signal Vout 1014. Currents are established by a current source 1008.

FIG. 51 is a circuit diagram of a low voltage op amp 1000 that corresponds to the block diagram of the op amp 1000 in FIG. 50. The op amp 1000 includes an input stage 1002, an output stage 1004 with a drive stage 1006, and a current source 1008. The op amp 1000 receives input signals Vin- 1010 and Vin+ 1012, and outputs an output signal Vout 1014.

The input signals Vin- 1010 and Vin+ 1012 are provided to bases of NPN bipolar transistors 1210 and 1212, respectively, in the input stage 1002. The input stage 1002 also includes p-channel MOSFET's 1202 and 1204. The NPN transistors 1210 and 1212 control amount of current that flows through the p-channel MOSFET 1202 and the p-channel MOSFET 1204, respectively. The p-channel MOSFET's 1202 and 1204 can be PMOS devices. In addition, the input stage 1002 includes a resistor 1206 and a capacitor 1208 coupled in series between collectors of the NPN transistors 1210 and 1212.

The collectors of the NPN transistors 1210 and 1212 are coupled to drains of the p-channel MOSFET's 1202 and 1204, respectively. Sources of the p-channel MOSFET's 1202 and 1204 are coupled to a positive voltage supply bus Vpp 1200. Substrates of the MOSFET's 1202 and 1204 are also coupled to the positive voltage supply bus Vpp 1200. A gate of the p-channel MOSFET 1202 is coupled to the drain of the p-channel MOSFET 1202. Thus, the p-channel MOSFET is configured as a diode and current flows through the p-channel MOSFET 1202 and the NPN transistor 1210. The amount of this current is controlled by the voltage applied at the base of the NPN transistor 1210, i.e., the inverting input signal Vin- 1010. A gate of the p-channel MOSFET 1204 is also coupled to the gate and the drain of the p-channel MOSFET 1202.

Emitters of the NPN transistors 1210 and 1212 are coupled to the current source 1008, which is used to provide currents that flow through the NPN transistors 1210 and 1212, respectively. The currents through each NPN transistor 1210 and 1212 is controlled by the voltage applied at its respective basis 1210 and 1212, respectively.

37367/CAG/B600

Since the current controlled by the current source 1008 is substantially constant, the sum of the currents flowing through the NPN transistors 1210 and 1212 are substantially constant as well. Thus, the ratio of the currents flowing through the NPN transistors 1210 and 1212 is determined by the ratio of the respective voltages of the input signals Vin- 1010 and Vin+ 1012. The drain of the p-channel MOSFET 1204 is coupled to the output stage 1004 and is provided as the output Vout 1014 through a filtering capacitor 1222 and the output stage 1004. The filtering capacitor 1222 operates as a high pass filter.

The current source 1008 includes n-channel MOSFET's 1216, 1218 and 1220. The n-channel MOSFET's 1216, 1218 and 1220 may be VMOS devices. The emitters of the NPN transistors 1210 and 1212 in the input stage 1002 are coupled to a drain of the n-channel MOSFET 1218. A drain of the n-channel MOSFET 1216 is coupled to the positive voltage supply bus Vpp 1200 through a resistor 1214. The drain and a gate of the n-channel MOSFET 1216 are coupled to each other.

Sources of the n-channel MOSFET's 1216, 1218 and 1220 are coupled to a negative voltage supply bus Vnn 1238. Thus, the n-channel MOSFET 1216 is configured as a diode, and current passes from the positive voltage bus Vpp 1200 through the resistor 1214 and the n-channel MOSFET 1216 to the negative voltage supply bus Vnn 1238, thereby fixing the gate voltage of the n-channel MOSFET 1216. Substrates of the n-channel MOSFET's 1216, 1218 and 1220 are coupled to a substrate voltage 1240.

The gate of the n-channel MOSFET 1216 is also coupled to gates of the n-channel MOSFET's 1218 and 1220, thereby fixing the gate voltage of each n-channel MOSFET in the current source 1008. Thus, the n-channel MOSFET 1216 is coupled in a current mirror configuration with the n-channel MOSFET's 1218 and 1220. In this current mirror configuration, the current through each of the n-channel MOSFET's 1218 and 1220 are similar in magnitude to the current through the n-channel MOSFET 1216 provided that the n-channel MOSFET's 1218 and 1220 have similar dimensions to the n-channel MOSFET 1216 and similar voltages are applied to their drains as voltage applied at the drain of the n-channel MOSFET 1216.

The output stage 1004 includes a p-channel MOSFET 1224 and the driver stage 1006. A source and a substrate of the p-channel MOSFET 1224 is coupled to the positive voltage supply bus Vpp 1200. A gate of the p-channel MOSFET 1224 is coupled to the drain of the p-channel MOSFET 1204 in the input stage and a first terminal of the filtering capacitor 1222. A drain of the p-channel MOSFET 1224 is coupled to the driver stage 1006. The driver stage 1006 is coupled to a source of the n-channel MOSFET 1220 in the current source 1008. Thus, a

37367/CAG/B600

current from the positive voltage supply bus Vpp 1200 that flows through the p-channel MOSFET 1224 and provided to the driver stage 1006 is controlled by the voltage at the drain of the p-channel MOSFET 1204.

The driver stage 1006 includes NPN bipolar transistors 1226, 1230 and PNP bipolar transistors 1228, 1236. A collector of the NPN transistor 1226 is coupled to the drain of the p-channel MOSFET 1224 in the output stage 1004. The collector of the NPN transistor 1226 is also coupled to a base of the NPN transistor 1226. Thus, the NPN transistor 1226 is configured as a diode. An emitter of the NPN transistor 1226 is coupled to an emitter of the PNP transistor 1228. A base and a collector of the PNP transistor 1228 is coupled to each other. Thus, the NPN transistor 1228 is also configured as a diode. The collector of the PNP transistor 1228 is also coupled to the drain of the n-channel MOSFET 1220 in the current source 1008.

The drain of the p-channel MOSFET 1224 and the collector of the NPN transistor 1226 are also coupled to a base of the NPN transistor 1230. A collector of the NPN transistor 1230 is coupled to the positive voltage supply bus Vpp 1200. An emitter of the NPN transistor 1230 is coupled to the output signal Vout 1014 through a resistor 1232.

An emitter of the PNP transistor 1236 is coupled to the output signal Vout 1014 through a resistor 1234. A base of the PNP transistor 1236 is coupled to the collector of the PNP transistor 1228 and the drain of the n-channel MOSFET 1220. A collector of the PNP transistor 1236 is coupled to a negative voltage supply bus Vnn 1238.

The transistors 1226 and 1228, configured as diodes, are used as bias compensating diodes. Therefore, the driver stage 1006 includes a bias compensation circuit to reduce cross over distortion. With the bias compensation of the drive stage 1006, the bias point may be stabilized, along with emitter degeneration, over dynamic operating conditions such as temperature.

The bipolar transistors 1230 and 1236 are used as power drivers in the driver stage. The bipolar transistors 1230 and 1236 operate as a Class A-B push-pull amplifier. When the voltage at the drain of the p-channel MOSFET 1204 is sufficiently low, e.g., lower than a threshold voltage, V_T, the p-channel MOSFET 1224 allows a current to flow through it, and voltage at the drain of the p-channel MOSFET 1224 approaches the positive power supply voltage Vpp.

As voltage at the drain of the p-channel MOSFET 1224 increases, the V_{BE} (base-to-emitter voltage) of the NPN transistor 1230 increases since the drain of the p-channel MOSFET 1224 is coupled to the base of the NPN transistor 1230. As the V_{BE} increases, the coupling

37367/CAG/B600

between the positive voltage supply bus Vpp 1200 and the output Vout 1014 strengthens, and therefore, the output Vout 1014 tends to be driven up stronger toward the positive power supply voltage, Vpp.

Meanwhile, voltage at the collector of the PNP transistor 1228 tends to increase as well, thus tending to turn off the PNP transistor 1236 as the VBE increases since the collector of the PNP transistor 1228 is coupled to the base of the PNP transistor 1236. Therefore, the output Vout 1014 tends not to be driven down as strongly toward the negative power supply voltage, Vnn.

On the other hand, as voltage applied at the gate of the p-channel MOSFET 1224 decreases, the p-channel MOSFET 1224 tends to turn off, and the positive power supply Vpp tends not to be propagated to the drain of the p-channel MOSFET 1224. Since the drain of the p-channel MOSFET 1224 is coupled to the base of the NPN transistor 1230, the NPN transistor 1230 tends to turn off, and the output Vout 1014 does not tend to be driven up toward the positive supply voltage Vpp.

At the same time, the negative supply voltage Vnn is propagated to the drain of the n-channel MOSFET 1220, and therefore applied at the base of the PNP transistor 1236. Thus, the PNP transistor 1236 tends to drive the output Vout 1014 down toward the negative supply voltage Vnn.

FIG. 52 is a circuit diagram of a high voltage op amp 1000 that corresponds to the block diagram of the op amp 1000 in FIG. 50. The op amp 1000 includes an input stage 1002, an output stage 1004 with a drive stage 1006, and a current source 1008. The op amp 1000 receives input signals Vin- 1010 and Vin+ 1012, and outputs an output signal Vout 1014.

The input stage 1002 includes p-channel MOSFET's 1302 and 1304. The p-channel MOSFET's 1302 and 1304 can be PMOS devices. The input stage 1002 also includes NPN bipolar transistors 1306, 1308, 1310 and 1312. The NPN bipolar transistors 1306 and 1308 are input transistors that receive inverting and non-inverting inputs Vin- and Vin+, respectively.

Sources of the p-channel MOSFET's 1302 and 1304 are coupled to a positive voltage supply bus Vpp 1300. Gates of the p-channel MOSFET's 1302 and 1304 are coupled to each other. The gate of the p-channel MOSFET 1302 is also coupled to a drain of the p-channel MOSFET 1302. Thus, the p-channel MOSFET 1302 is configured as a diode.

A drain of the p-channel MOSFET 1302 is also coupled to collectors of the NPN transistors 1306 and 1310. A base of the NPN transistor 1306 receives an inverting input signal Vin- 1010. An emitter of the NPN transistor 1306 is coupled to a base of the NPN transistor 1310. A drain of the p-channel MOSFET 1304 is coupled to a collector of the NPN transistor 1308 and a collector of the NPN transistor 1312. The drain of the p-channel MOSFET 1304 is also coupled to the output stage 1004. A base of the NPN transistor 1308 is coupled to a non-inverting input signal Vin+. An emitter of the NPN transistor 1308 is coupled to a base of the NPN transistor 1312. Emitters of the NPN transistors 1310 and 1312 are coupled to each other and also coupled to the current source 1008.

The current drawn by the current source 1008 from the emitters of the NPN transistors 1310 and 1312 is substantially constant. Therefore, the sum of currents flowing through the NPN transistors 1310 and 1312 are substantially constant as well. The ratio between the currents flowing through the NPN transistor 1310 and the NPN transistor 1312, respectively, is controlled by the ratio of input voltages Vin- 1010 and Vin+ 1012.

The current source 1008 includes n-channel MOSFET's 1316, 1318, 1320, 1330 and 1332. These MOSFET's are configured either as a diode or a current mirror, and are used as current source for the input stage 1002 and the output stage 1004. The n-channel MOSFET's 1316, 1318, 1320, 1330 and 1332 can be VMOS devices. Substrates of the n-channel MOSFET's 1316, 1318 and 1320 are coupled to a substrate voltage 1374. Sources of the n-channel MOSFET's 1316, 1318 and 1320 are coupled to the negative voltage supply bus Vnn 1372.

A drain and a gate of the n-channel MOSFET 1316 are coupled to each other. Thus, the n-channel MOSFET 1316 is configured as a diode. The drain of the n-channel MOSFET 1316 is also coupled to a positive voltage source 1370 through a resistor 1314. Thus the current flowing through the n-channel MOSFET 1316 is controlled by the resistance value of the resistor 1314.

The drain of the n-channel MOSFET 1318 is coupled to the emitters of the NPN transistors 1310 and 1312 in the input stage 1002. Gates of the n-channel MOSFET's 1318 and 1320 are coupled to the gate of the n-channel MOSFET 1316. Thus, the n-channel MOSFET's 1318 and 1320 are configured as current mirrors to the n-channel MOSFET 1316. Therefore, currents flowing through the n-channel MOSFET's 1318 and 1320 is similar in magnitude to the current flowing through the n-channel MOSFET 1316 provided that the n-channel MOSFET's 1318 and 1320 have similar dimensions as the n-channel MOSFET 1316 and voltages at the

drains of the n-channel MOSFET's 1318 and 1320 are similar to the voltage at the drain of the n-channel MOSFET 1316.

The current source 1008 also includes PNP bipolar transistors 1326 and 1328. Emitters of the PNP transistors 1326 and 1328 are coupled to a positive voltage supply bus Vpp 1300 through a bias resistor 1322 and a bias resistor 1324, respectively. A base and a collector of the PNP transistor 1326 are coupled to each other. Thus, the PNP transistor 1326 is configured as a diode. The collector of the PNP transistor 1326 is also coupled to a drain of the n-channel MOSFET 1320. The base of the PNP transistor 1326 is also coupled to a base of the PNP transistor 1328. Thus, the PNP transistor 1328 is configured as a current mirror to the PNP transistor 1326.

A collector of the PNP transistor 1328 is coupled to a drain and a gate of the n-channel MOSFET 1330. Since the drain and the gate of the n-channel MOSFET 1330 are coupled to each other, the n-channel MOSFET 1330 is configured as a diode. The gate of the n-channel MOSFET 1330 is also coupled to a gate of the n-channel MOSFET 1332. Thus, the n-channel MOSFET 1332 is configured as a current mirror to the n-channel MOSFET 1330. Sources and substrates of the n-channel MOSFET's 1330 and 1332 are coupled to the negative voltage supply bus Vnn 1372. A drain of the n-channel MOSFET 1332 is coupled to the output stage 1004.

The output stage 1004 includes PNP bipolar transistors 1338, 1342 and a driver stage 1006. A base of the PNP transistor 1338 is coupled to the drain of the p-channel MOSFET 1304 in the input stage 1002. The voltage at the drain of the p-channel MOSFET 1304 is provided as an output signal Vout 1014 through a current-limiting resistor 1334 and a filtering capacitor 1336 in series. The filtering capacitor 1336 operates as a high pass filter. A collector of the PNP transistor 1338 is coupled to a negative voltage supply 1340, e.g., ground. An emitter of the PNP transistor 1338 is coupled to a base of the PNP transistor 1342. An emitter of the PNP transistor 1342 is coupled to the positive voltage supply bus Vpp 1300. A collector of the PNP transistor 1342 is coupled to the driver stage 1006.

The driver stage 1006 includes NPN bipolar transistors 1344, 1348, PNP bipolar transistors 1346, 1350, 1352, and n-channel MOSFET's 1354, 1364. A collector of the NPN transistor 1344 is coupled to the collector of the PNP transistor 1342 of the output stage 1004. The collector of the NPN transistor 1344 is also coupled to a base of the NPN transistor 1344 and a base of the NPN transistor 1348. Thus, the NPN transistor 1344 is configured as a diode. An emitter of the NPN transistor 1344 is coupled to an emitter of the PNP transistor 1346. A base and a collector of the PNP transistor 1346 are coupled to each other. Thus, the PNP transistor 1346 is configured as a diode. A collector of the PNP transistor 1346 is coupled to the drain of

37367/CAG/B600

the n-channel MOSFET 1332 in the current source 1008. Therefore, the NPN transistor 1344 and the PNP transistor 1346 are configured as bias compensating diodes. The bias compensating diodes are used for bias control and stability, and the use of these diodes results in enhanced performance such as low distortion, low quiescent power dissipation and dynamic control of bias point.

The collector of the NPN transistor 1348 is coupled to a base of the PNP transistor 1350. An emitter of the NPN transistor 1348 is coupled to a source of the n-channel MOSFET 1354. An emitter of the PNP transistor 1350 is coupled to the positive voltage supply bus Vpp 1300. A collector of the PNP transistor 1350 is coupled to a gate of the n-channel MOSFET 1354. The collector of the PNP transistor 1350 is also coupled to a source of the n-channel MOSFET 1354 through a resistor 1356 and a Zener diode 1358 in parallel. A drain of the n-channel MOSFET 1354 is coupled to the positive voltage supply bus Vpp 1300. A substrate of the n-channel MOSFET 1354 is coupled to the negative voltage supply bus Vnn 1372. The source of the n-channel MOSFET 1354 is coupled to the output signal Vout 1014 through a current-limiting resistor 1360.

An emitter of the PNP transistor 1352 is coupled to the output signal Vout 1014 through a current-limiting resistor 1362. A base of the PNP transistor 1352 is coupled to the collector of the PNP transistor 1346. A collector of the PNP transistor 1352 is coupled to the negative voltage supply bus Vnn 1372 through a Zener diode 1366 and a resistor 1368 in parallel. The collector of the PNP transistor 1352 is also coupled to a gate of the n-channel MOSFET 1364. A drain of the n-channel MOSFET 1364 is coupled to the emitter of the PNP transistor 1352, and also, through the resistor 1362, coupled to the output signal Vout 1014. A substrate of the n-channel MOSFET 1364 is coupled to the negative voltage supply bus Vnn 1372.

As the voltage at the base of the PNP transistor 1338 decreases, the base-to-emitter voltage (V_{BE}) decreases. As the V_{BE} falls below a threshold voltage (Vth), the PNP transistor 1338 turns on, and the negative supply voltage 1340 pulls down the voltage at the base of the PNP transistor 1342, turning on the PNP transistor 1342. When the PNP transistor 1342 turns on, the positive voltage supply Vpp pulls up the voltage at the base of the NPN transistor 1348, thus turning on the NPN transistor 1348.

As the NPN transistor 1348 turns on, the V_{BE} of the PNP transistor 1350 tends to decrease, turning on the PNP transistor 1350 to apply the positive supply voltage Vpp at the gate of the n-channel MOSFET 1354. When the n-channel MOSFET 1354 is turned on, the n-channel MOSFET 1354 tends to drive the output Vout 1014 toward the positive supply voltage Vpp. Meanwhile, the positive supply voltage Vpp also tends to pull up the base of the PNP transistor

1352 through the NPN transistor 1344 and the PNP transistor 1346 which are configured as bias compensating diodes. Therefore, the PNP transistor 1352 is not turned on, and does not drive the output Vout 1014 toward the negative voltage supply Vnn.

When the voltage applied at the base of the PNP transistor 1338 is sufficiently high such that V_{BE} > Vth, the PNP transistor 1338 does not turn on, and the PNP transistor 1342 does not turn on. No substantial current flow through the NPN transistor 1344 and the PNP transistor 1346. The NPN transistor 1348, the PNP transistor 1350 and the n-channel MOSFET 1354 do not turn on. Therefore, the output Vout 1014 is not pulled up toward the positive supply voltage Vpp.

At the same time, since the n-channel MOSFET 1332 is turned on, the negative supply voltage Vnn, e.g., ground, is propagated to the drain of the n-channel MOSFET 1332 and thus applied at the base of the PNP transistor 1352, turning on the PNP transistor 1352. As the PNP transistor 1352 turns on, the gate of the n-channel MOSFET 1364 is pulled up, turning on the n-channel MOSFET 1364. As the n-channel MOSFET 1364 turns on, it drives the output Vout 1014 toward the negative supply voltage Vnn.

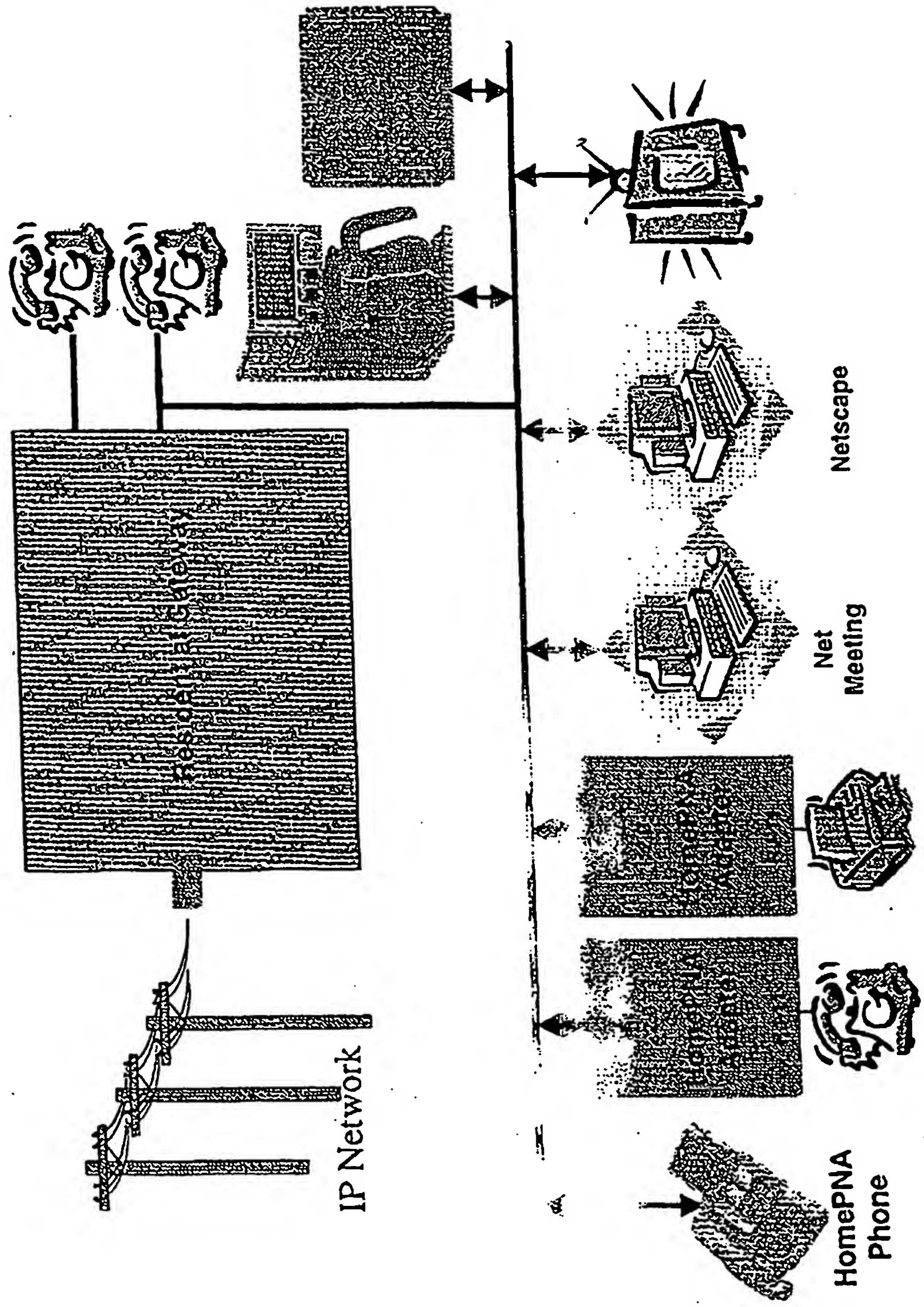


Figure 1

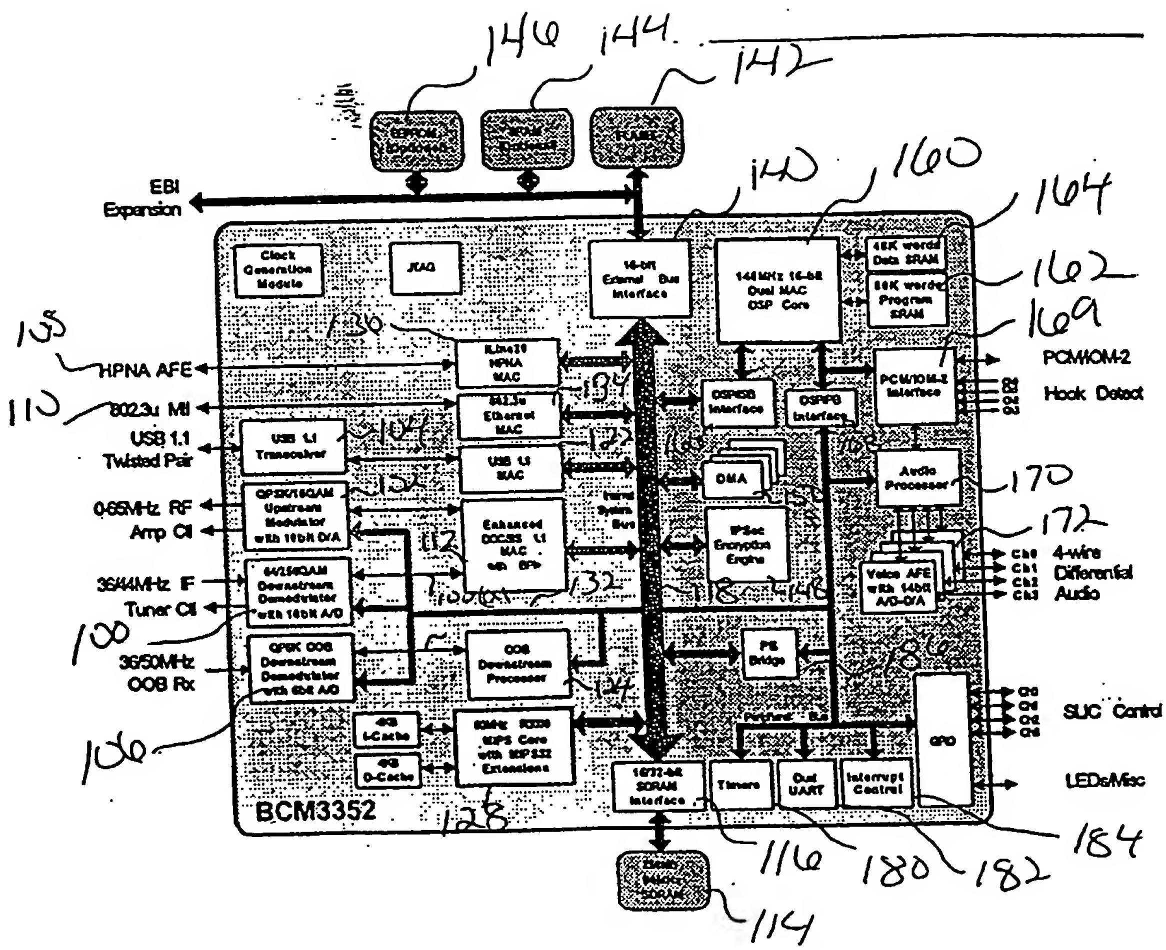

EBI Expansion
Clock Generation Module
JTAG
16-bit External Bus Interface
144MHz 16-bit Dual MAC DSP Core
48K words Data SRAM
80K words Program SRAM
HPNA AFE
HPNA MAC
802.3u MII
802.3u Ethernet MAC
USB 1.1 Twisted Pair
USB 1.1 Transceiver
USB 1.1 MAC
0-65MHz RF
Amp Ctl
QPSK/16QAM Upstream Modulator with 10bit D/A
Enhanced DOCSIS 1.1 MAC
36/44MHz IF
Tuner Ctl
64/256QAM Downstream Demodulator with 10bit A/D
36/50MHz OOB Rx
QPSK OOB Downstream Demodulator with 6bit A/D
OOB Downstream Processor
I-Cache
D-Cache
MIPS Core with MIPS32 Extensions
SDRAM Interface
BCM3352
DSPIO Interface
DSPPB Interface
PCM/IOM-2 Interface
PCM/IOM-2
Hook Detect
DMA
Audio Processor
IPSec Encryption Engine
Voice AFE with 14bit A/D-D/A
Ch0
Ch1
Ch2
Ch3
4-wire Differential Audio
PB Bridge
GPIO
SLIC Control
LEDs/Misc
Timers
Dual UART
Interrupt Control
146
144
142
140
160
164
162
169
108
110
130
134
104
122
124
112
132
100
106
128
116
180
182
184
114
170
172
148
118

FIG. 2

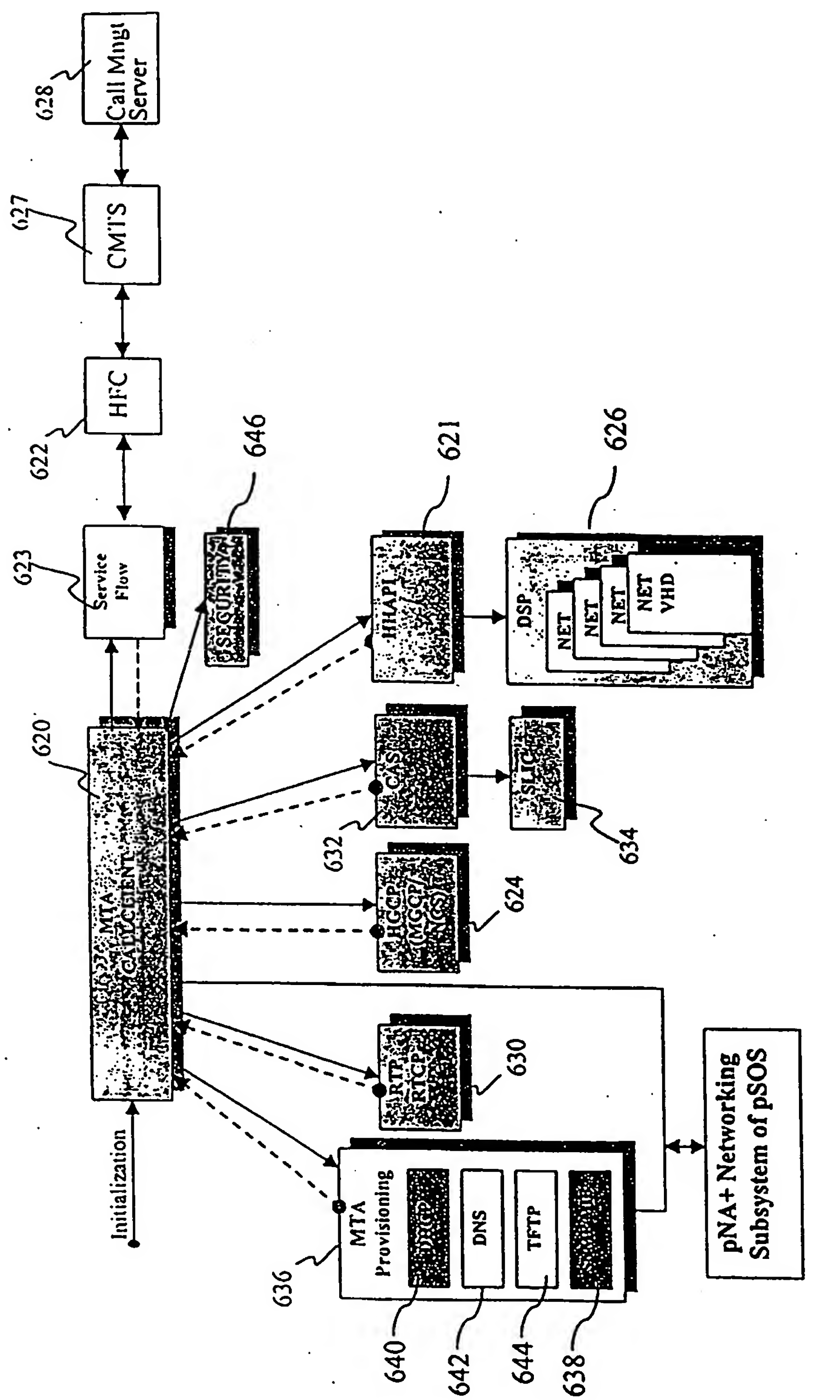

628
Call Mngt Server
627
CMTS
622
HFC
623
Service Flow
646
SECURITY
620
MTA
CALLCLIENT
621
HHAPI
626
DSP
NET
NET
NET
NET
VHD
632
CAS
SLIC
634
HGCP (MGCP/NCS)
624
RTP
RTCP
630
Initialization
636
MTA Provisioning
DHCP
DNS
TFTP
640
642
644
638
pNA+ Networking Subsystem of pSOS
Legend:
Callback functions
Direct function calls

FIG. 3

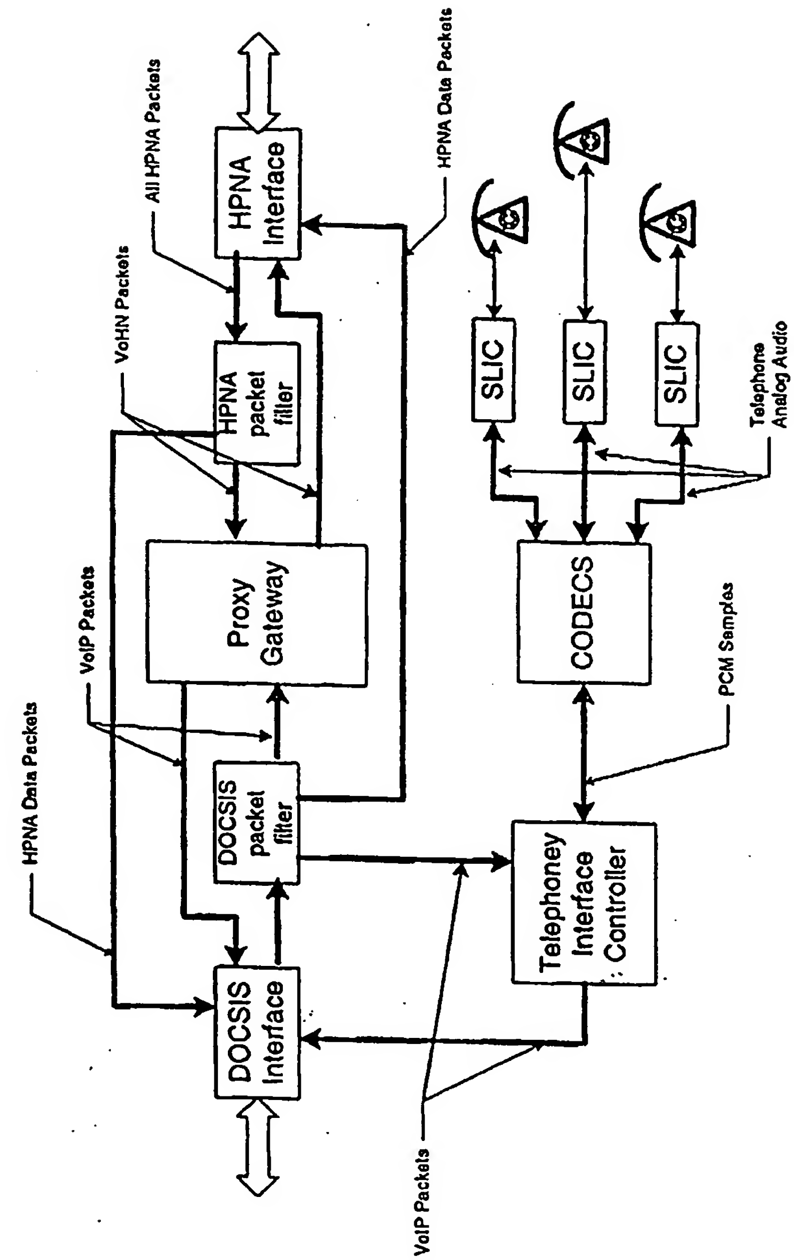

All HPNA Packets
HPNA Interface
HPNA Data Packets
VoHN Packets
HPNA packet filter
VoIP Packets
Proxy Gateway
HPNA Data Packets
DOCSIS packet filter
DOCSIS Interface
VoIP Packets
Telephoney Interface Controller
PCM Samples
CODECS
SLIC
SLIC
SLIC
Telephone Analog Audio

Figure 4

Full HPNA Telephone

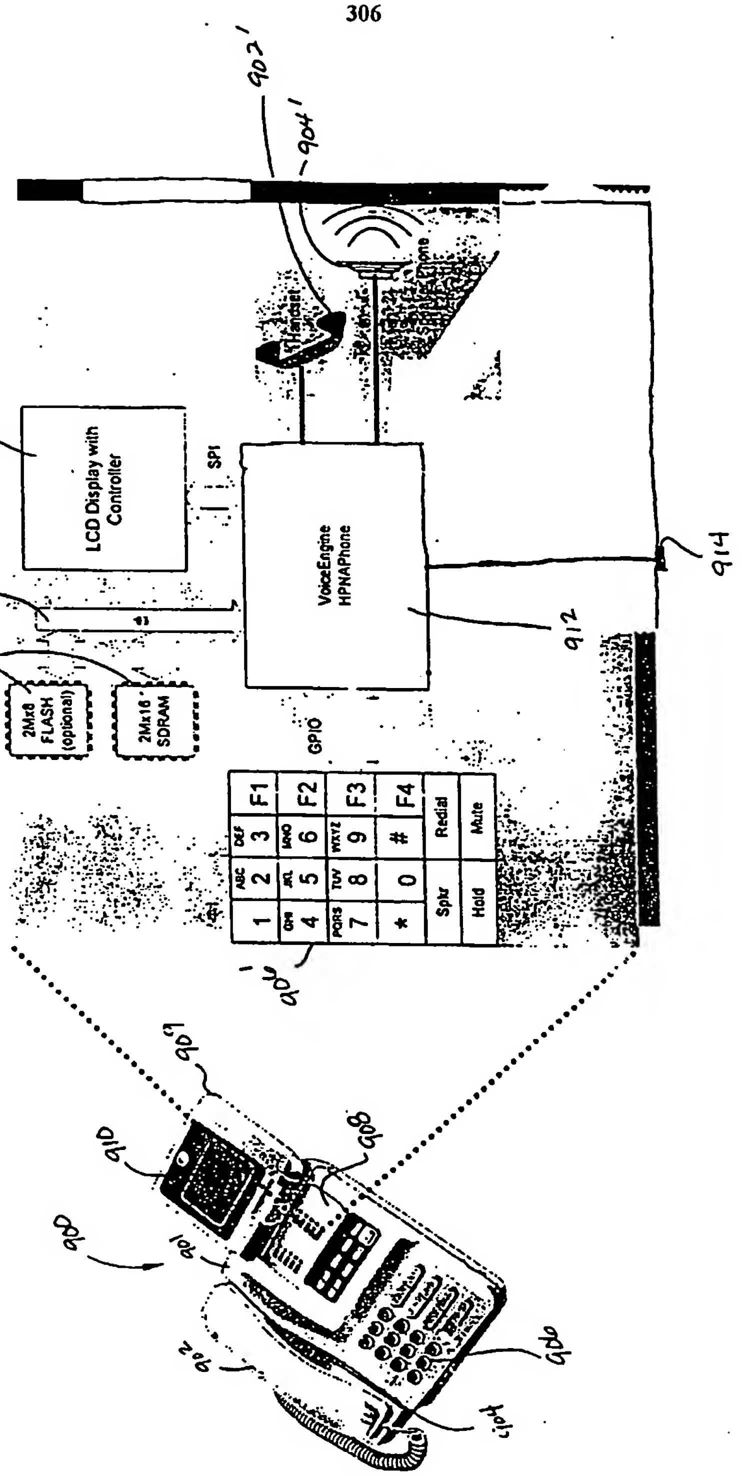


FIG. 5

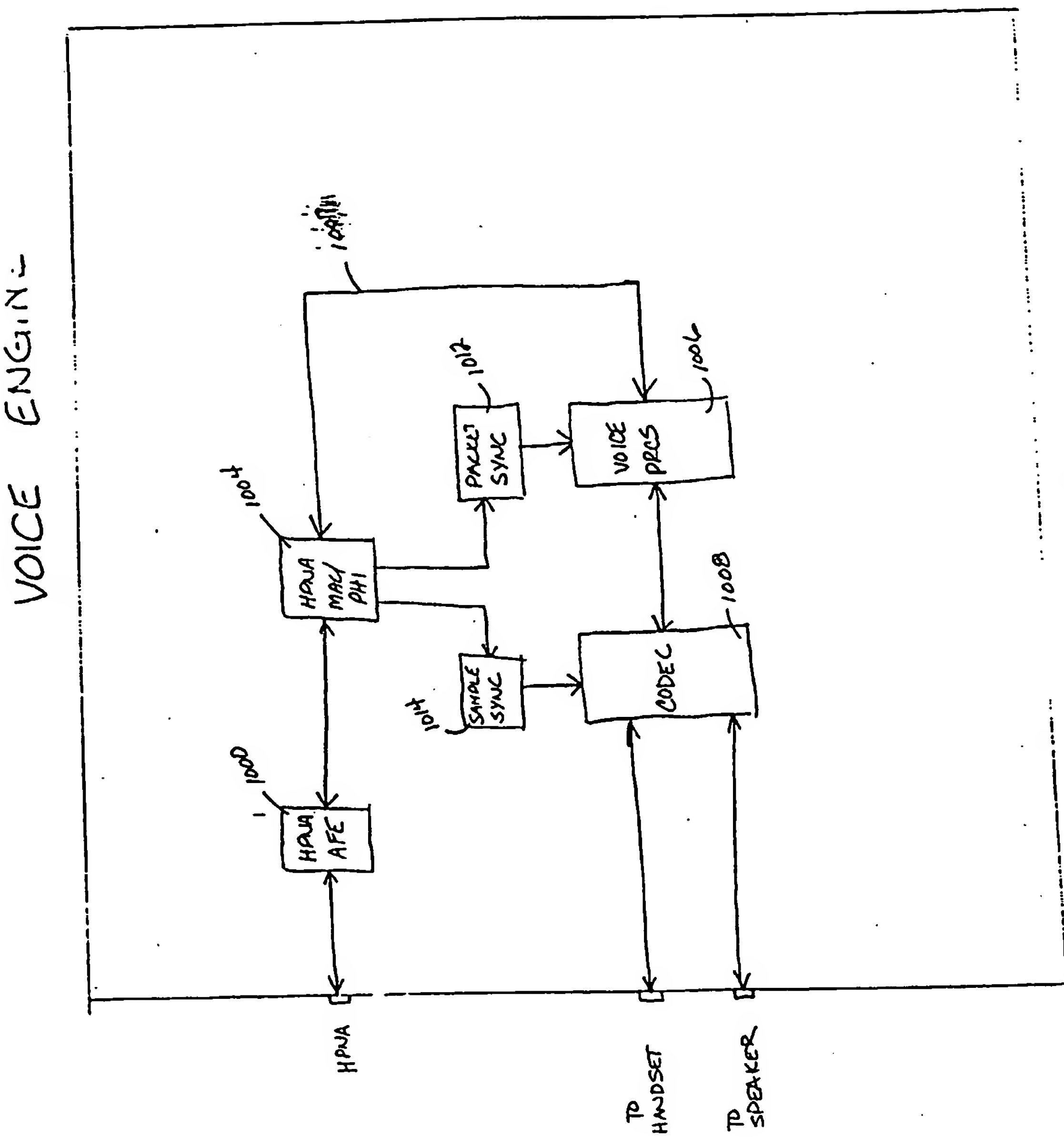

VOICE ENGINE
HPNA AFE
1000
HPNA MAC/PHI
1004
PACKET SYNC
1012
VOICE PROC
1006
CODEC
1008
SAMPLE SYNC
1014
HPNA
TP HANDSET
TP SPEAKER

FIG. 6

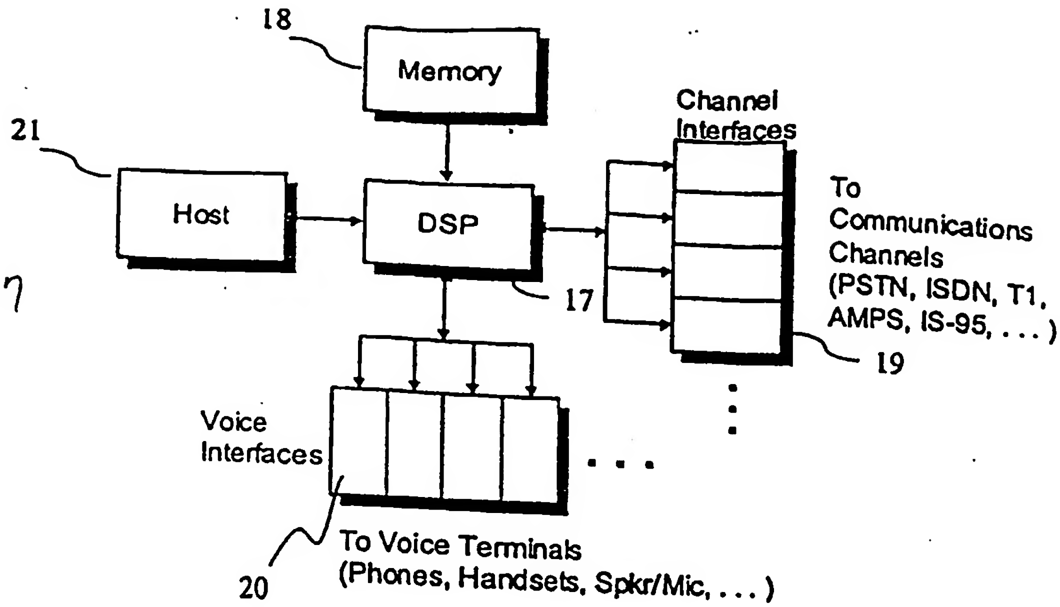

18
Memory
21
Host
DSP
17
Channel
Interfaces
To
Communications
Channels
(PSTN, ISDN, T1,
AMPS, IS-95, . . .)
19
Voice
Interfaces
20
To Voice Terminals
(Phones, Handsets, Spkr/Mic, . . .)

FIG. 7

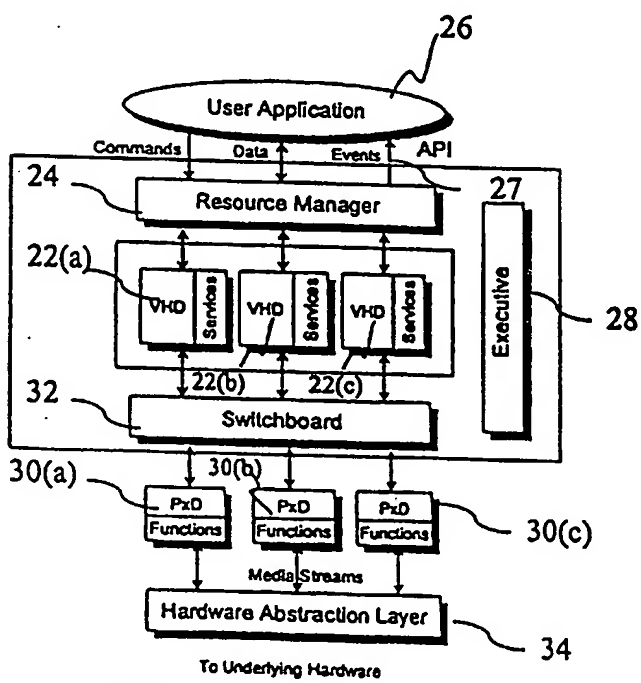

26
User Application
Commands
Data
Events
API
24
Resource Manager
27
22(a)
VHD
Services
22(b)
22(c)
Executive
28
32
Switchboard
30(a)
30(b)
PxD
Functions
30(c)
Media Streams
Hardware Abstraction Layer
34
To Underlying Hardware

FIG. 8

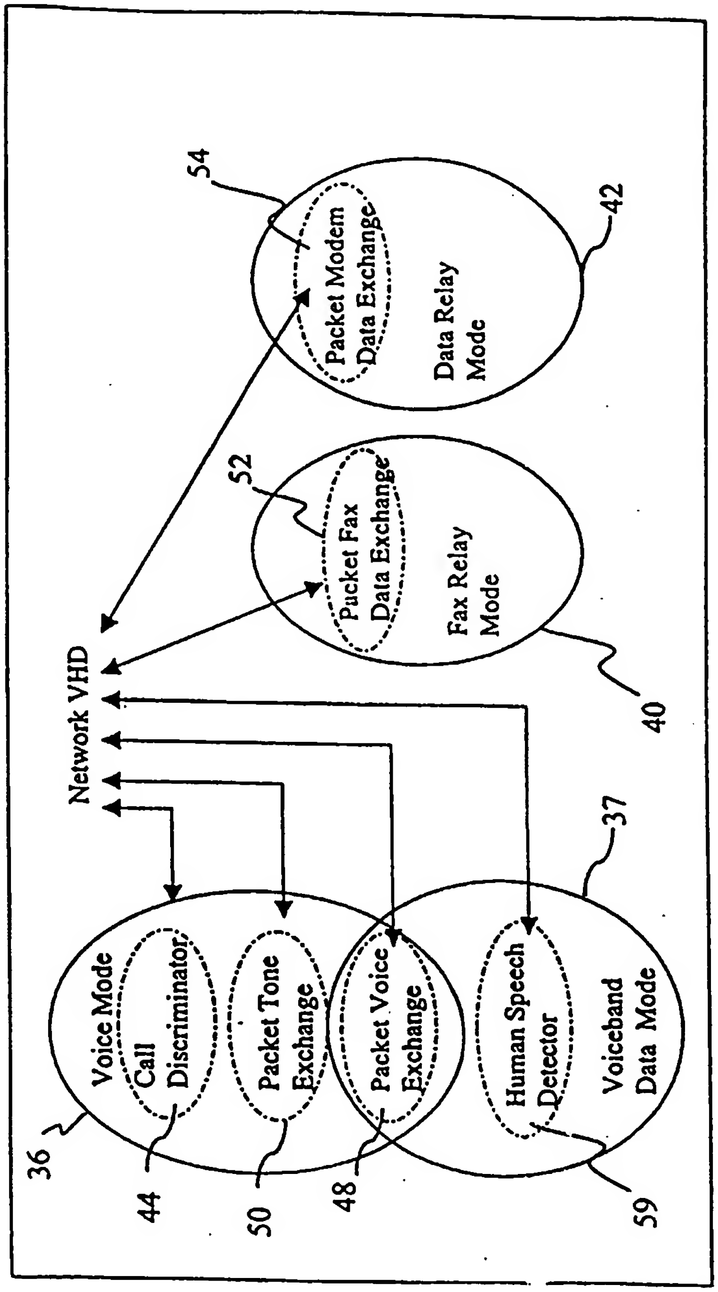

Network VHD
Voice Mode
Call Discriminator
Packet Tone Exchange
Packet Voice Exchange
Human Speech Detector
Voiceband Data Mode
Pucket Fax Data Exchange
Fax Relay Mode
Packet Modem Data Exchange
Data Relay Mode
36
44
50
48
59
37
40
52
54
42

FIG. 9

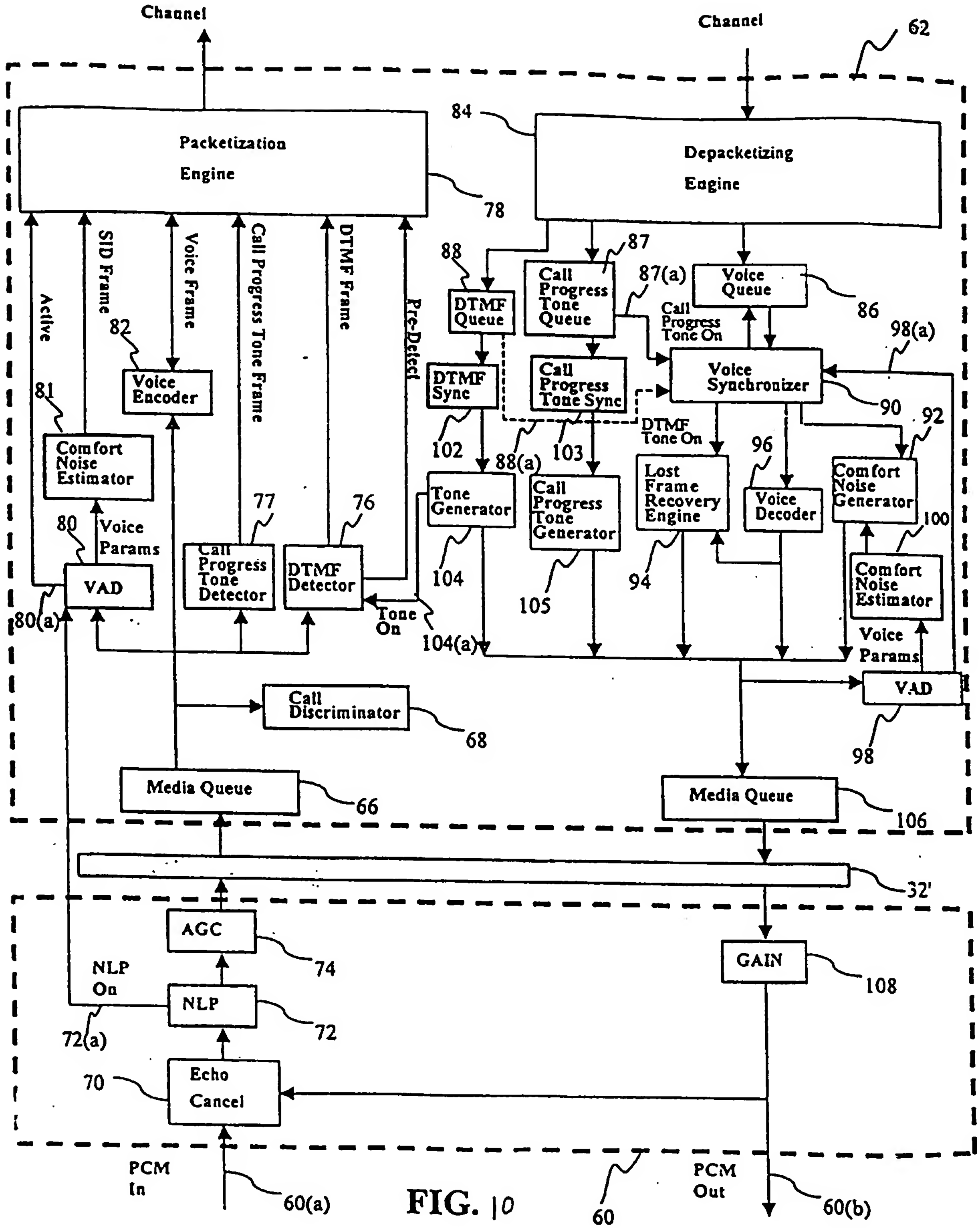

Channel
Channel
62
84
Packetization Engine
Depacketizing Engine
78
Active
SID Frame
Voice Frame
Call Progress Tone Frame
DTMF Frame
Pre-Detect
88
87
87(a)
DTMF Queue
Call Progress Tone Queue
Voice Queue
86
Call Progress Tone On
98(a)
82
Voice Encoder
81
DTMF Sync
Call Progress Tone Sync
Voice Synchronizer
90
92
DTMF Tone On
102
88(a)
103
96
Comfort Noise Estimator
Lost Frame Recovery Engine
Comfort Noise Generator
Tone Generator
Call Progress Tone Generator
Voice Decoder
100
77
76
80
Voice Params
Call Progress Tone Detector
DTMF Detector
104
105
94
Comfort Noise Estimator
VAD
80(a)
Tone On
104(a)
Voice Params
VAD
Call Discriminator
68
98
Media Queue
66
Media Queue
106
32'
AGC
74
GAIN
108
NLP On
NLP
72
72(a)
70
Echo Cancel
PCM In
60(a)
60
PCM Out
60(b)

FIG. 10

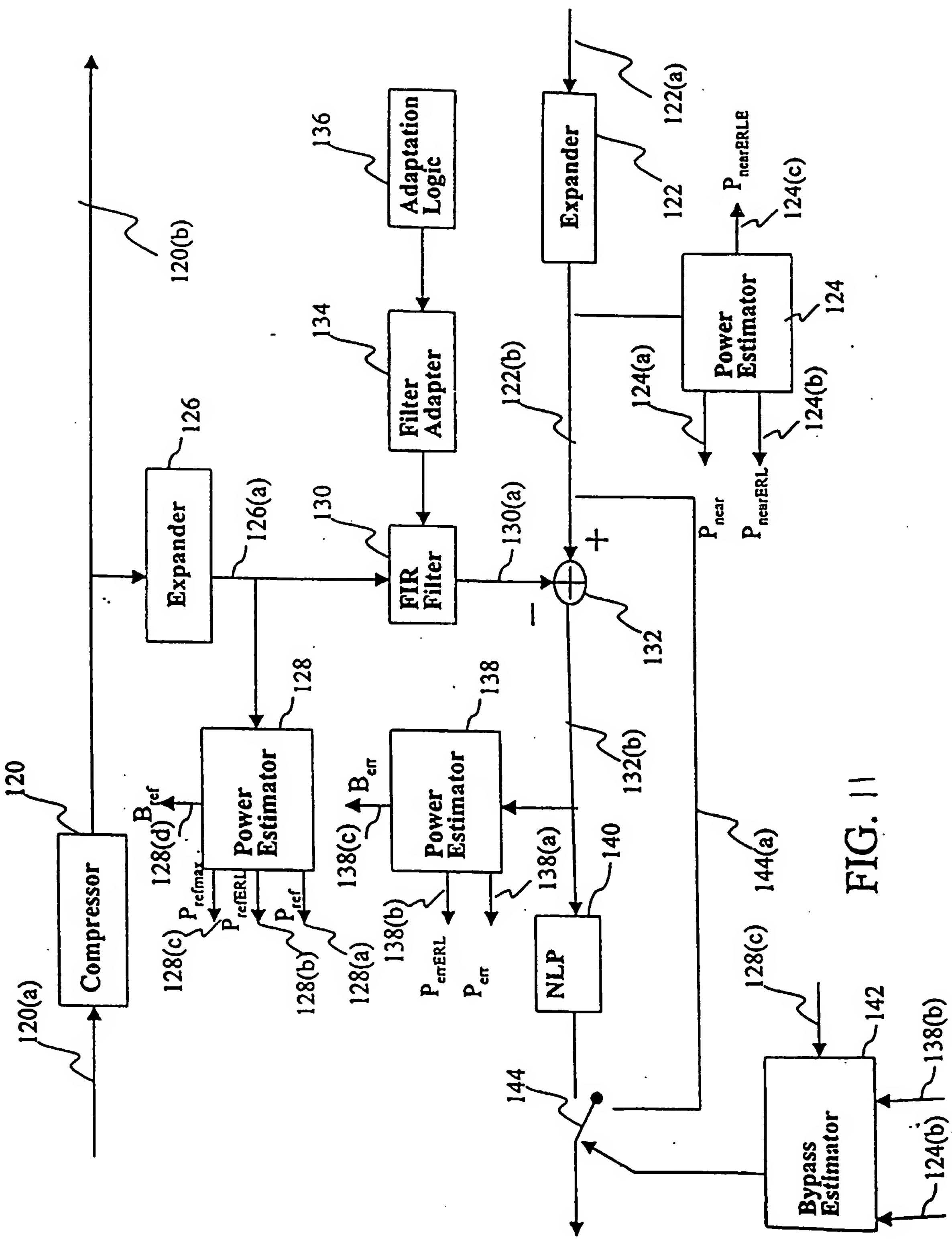
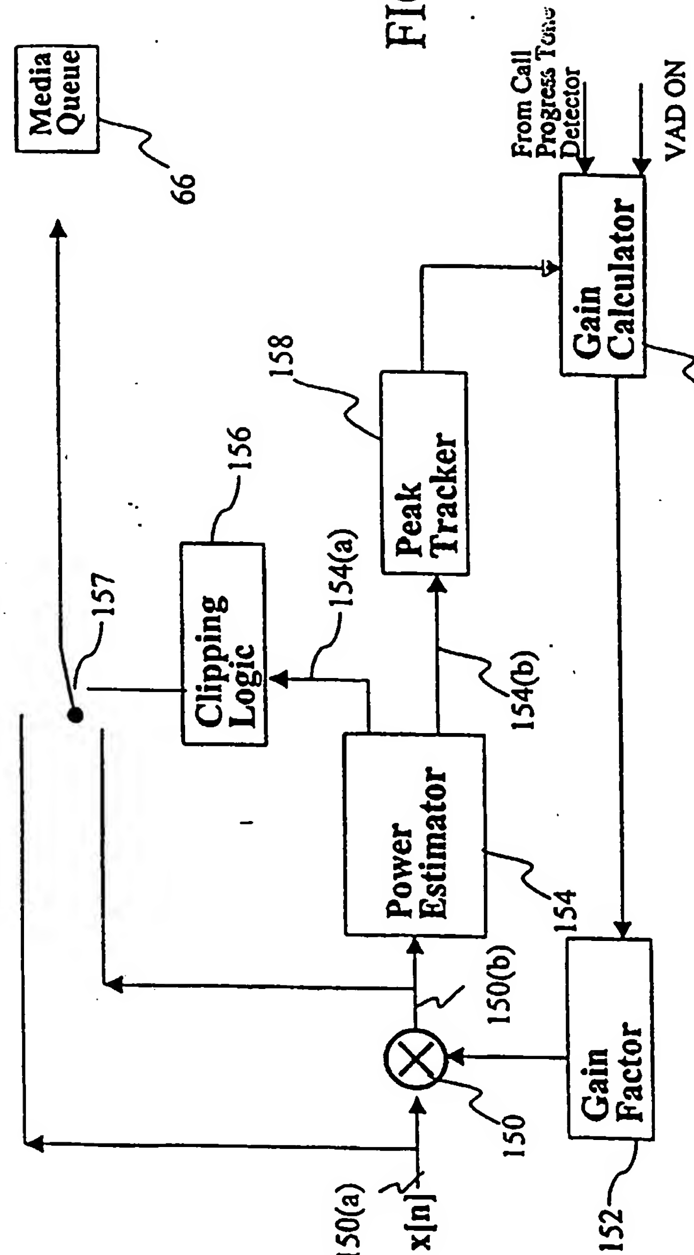

Compressor
120
120(a)
120(b)
Expander
126
126(a)
Power Estimator
128
128(a)
128(b)
128(c)
128(d)
B_{ref}
P_{refmax}
P_{refERL}
P_{ref}
Adaptation Logic
136
Filter Adapter
134
FIR Filter
130
130(a)
Power Estimator
138
138(a)
138(b)
138(c)
B_{err}
P_{errERL}
P_{err}
NLP
140
132
132(b)
+
-
Expander
122
122(a)
122(b)
Power Estimator
124
124(a)
124(b)
124(c)
$P_{nearERLE}$
P_{near}
$P_{nearERL}$
144
144(a)
Bypass Estimator
142
128(c)
138(b)
124(b)

FIG. 11

FIG. 12-A



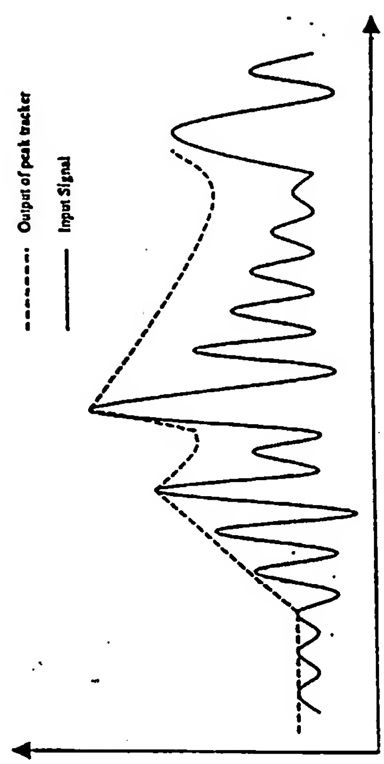


FIG. 12B

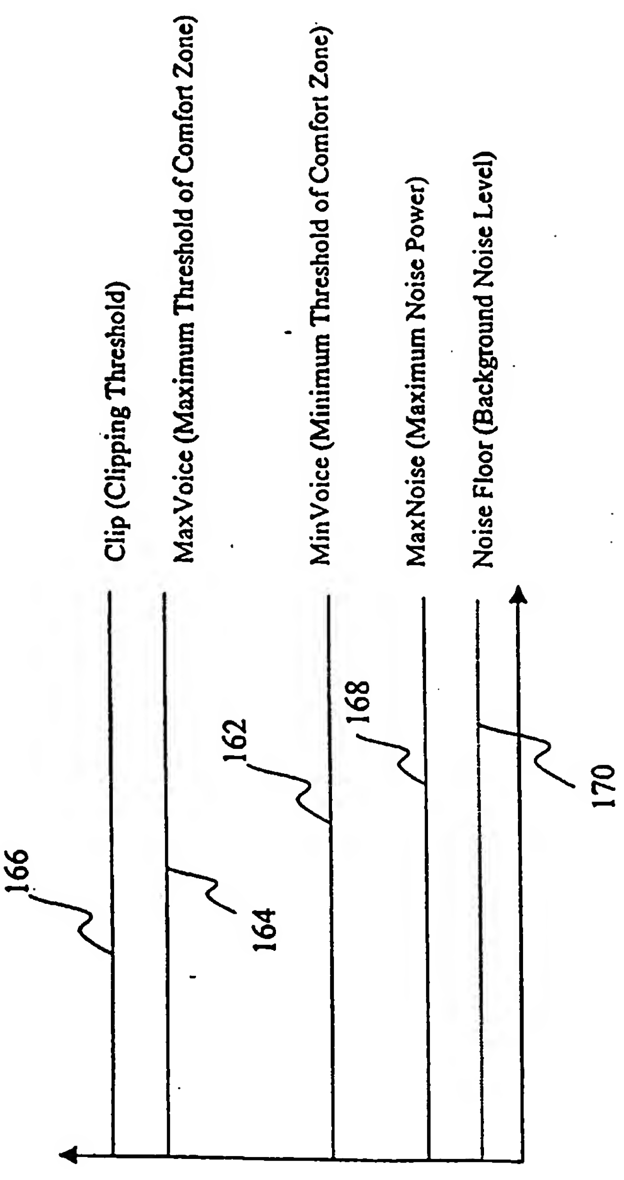

Clip (Clipping Threshold)
MaxVoice (Maximum Threshold of Comfort Zone)
MinVoice (Minimum Threshold of Comfort Zone)
MaxNoise (Maximum Noise Power)
Noise Floor (Background Noise Level)
166
164
162
168
170

FIG. 13

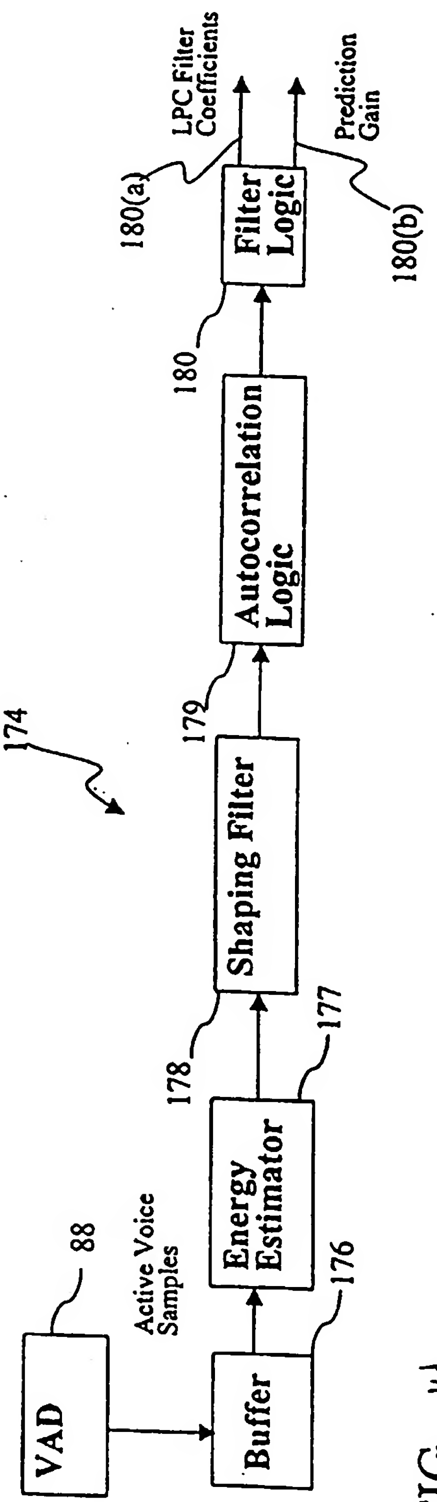

174
VAD
88
Active Voice Samples
Buffer
176
Energy Estimator
177
178
Shaping Filter
179
Autocorrelation Logic
180
Filter Logic
180(a)
180(b)
LPC Filter Coefficients
Prediction Gain

FIG. 14

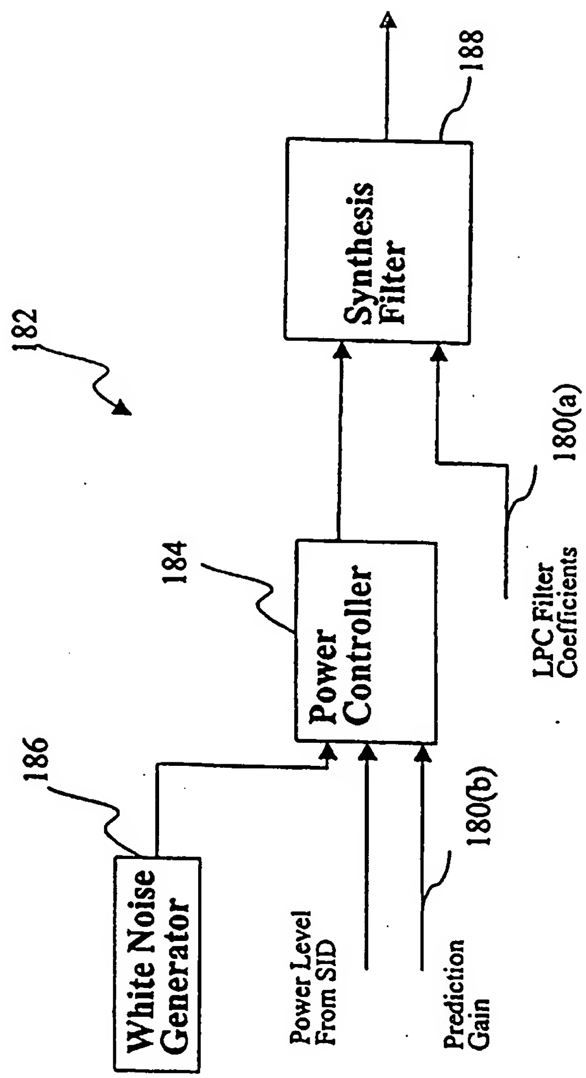

182
White Noise Generator
186
Power Level From SID
Prediction Gain
180(b)
Power Controller
184
LPC Filter Coefficients
180(a)
Synthesis Filter
188

FIG. 15

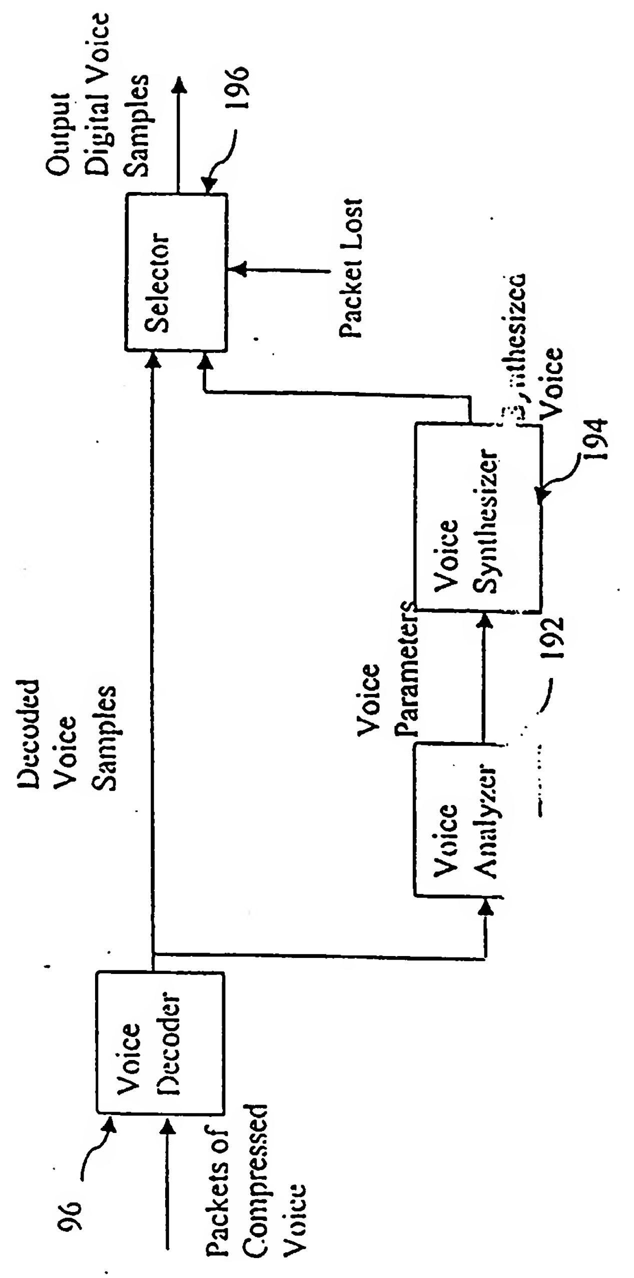

Output
Digital Voice
Samples
196
Selector
Packet Lost
Synthesized
Voice
194
Voice
Synthesizer
Voice
Parameters
192
Voice
Analyzer
Decoded
Voice
Samples
Voice
Decoder
96
Packets of
Compressed
Voice

FIG. 16

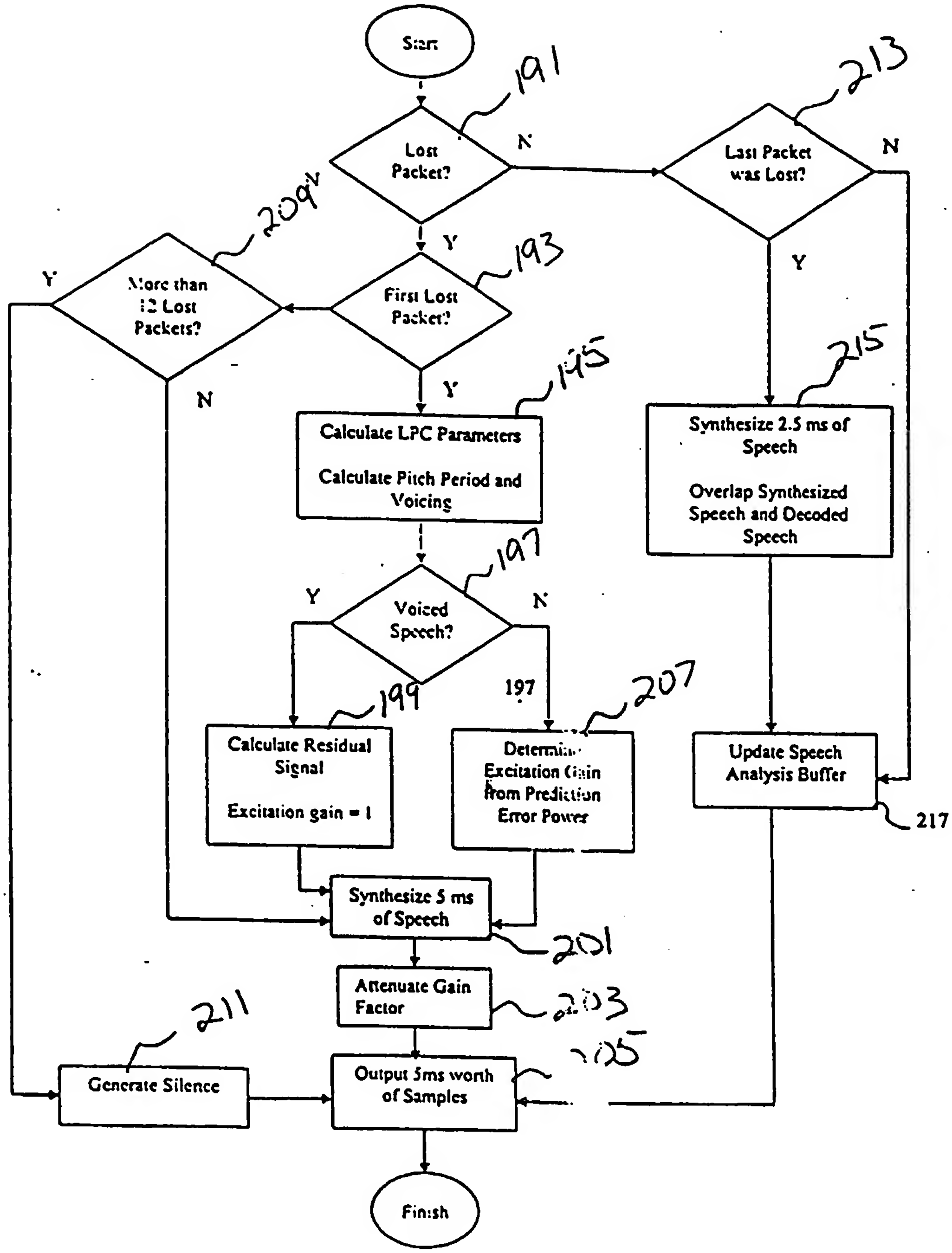

Start
Lost Packet?
N
Y
191
Last Packet was Lost?
N
Y
213
First Lost Packet?
193
More than 12 Lost Packets?
209
Y
N
Calculate LPC Parameters
Calculate Pitch Period and Voicing
195
Synthesize 2.5 ms of Speech
Overlap Synthesized Speech and Decoded Speech
215
Voiced Speech?
197
Y
N
Calculate Residual Signal
Excitation gain = 1
199
Determine Excitation Gain from Prediction Error Power
207
Update Speech Analysis Buffer
217
Synthesize 5 ms of Speech
201
Attenuate Gain Factor
203
Generate Silence
211
Output 5ms worth of Samples
205
Finish

FIG. 17A

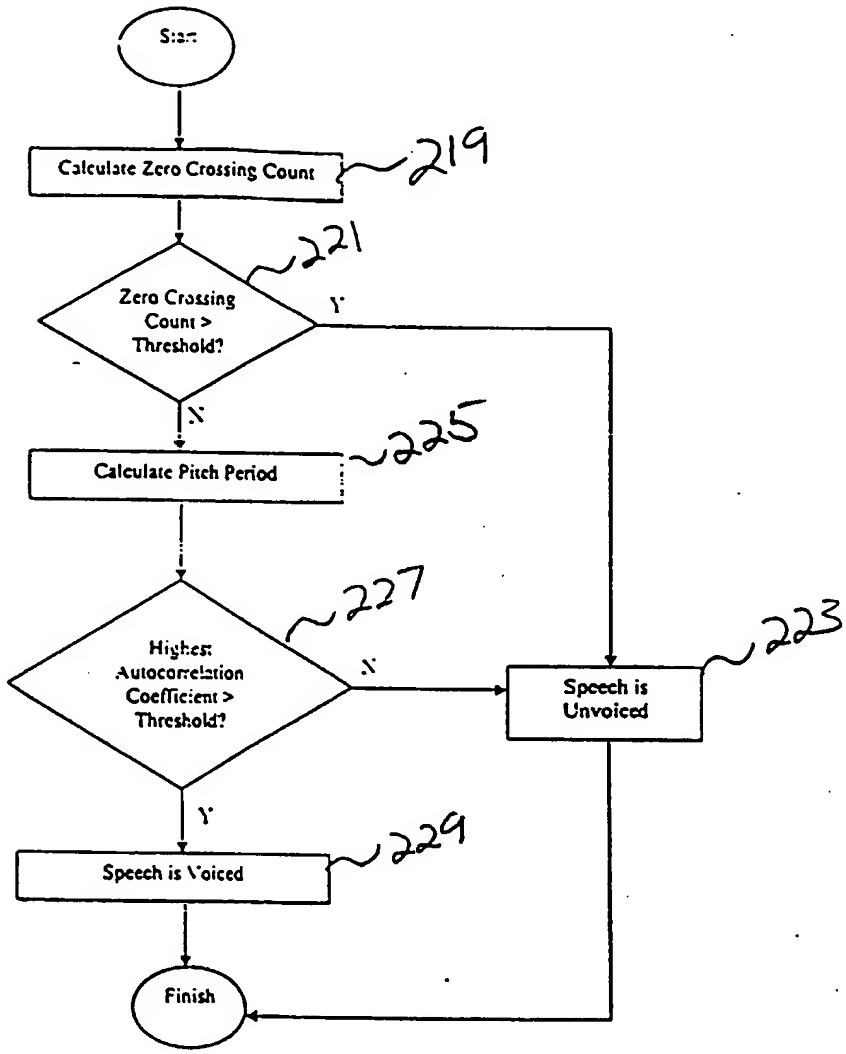

Start
Calculate Zero Crossing Count
219
221
Zero Crossing Count > Threshold?
Y
N
Calculate Pitch Period
225
227
Highest Autocorrelation Coefficient > Threshold?
N
Speech is Unvoiced
223
Y
Speech is Voiced
229
Finish

FIG. 17B

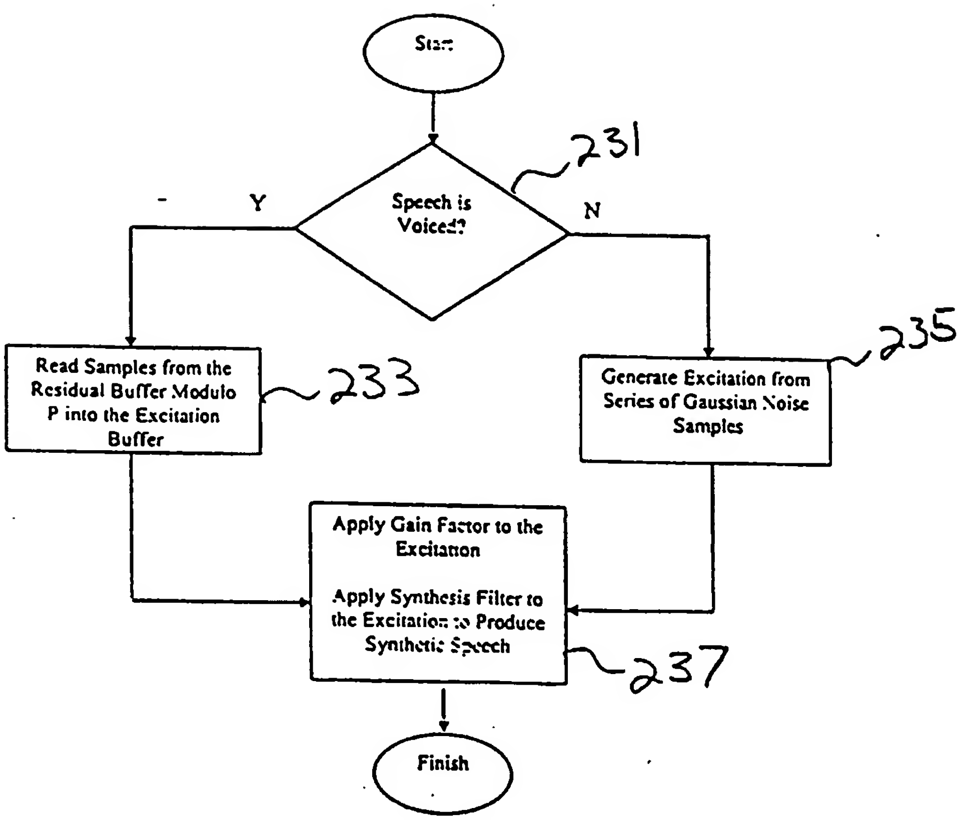

Start
Speech is Voiced?
231
Y
N
Read Samples from the Residual Buffer Modulo P into the Excitation Buffer
233
Generate Excitation from Series of Gaussian Noise Samples
235
Apply Gain Factor to the Excitation
Apply Synthesis Filter to the Excitation to Produce Synthetic Speech
237
Finish

FIG. 17C

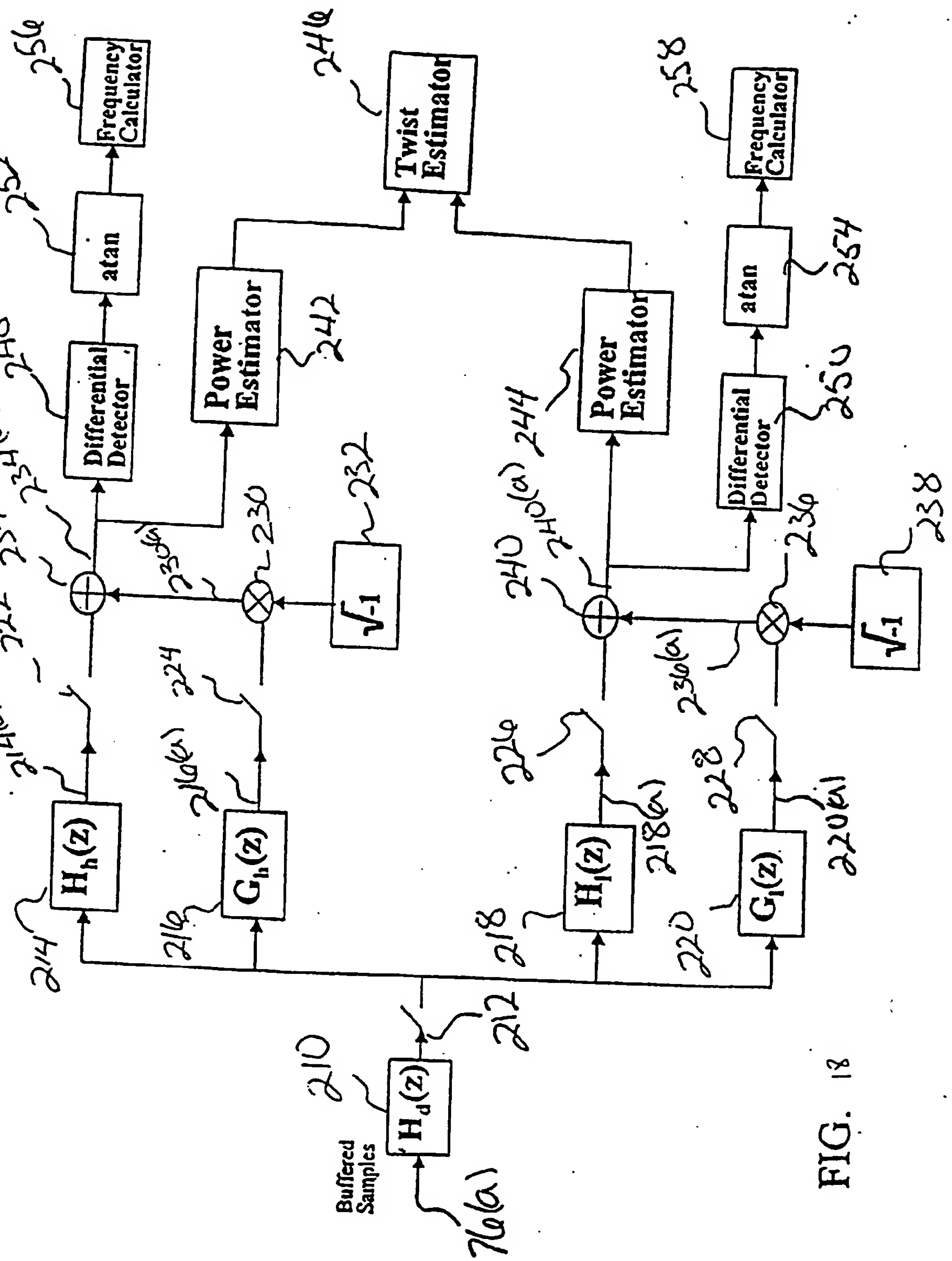

Buffered Samples
76(a)
$H_d(z)$
210
212
$H_h(z)$
214
214(a)
$G_h(z)$
216
216(a)
$H_l(z)$
218
218(a)
$G_l(z)$
220
220(a)
222
224
226
228
230
230(a)
$\sqrt{-1}$
232
234
234(a)
236
236(a)
$\sqrt{-1}$
238
240
240(a)
Differential Detector
248
atan
252
Frequency Calculator
256
Power Estimator
242
Twist Estimator
246
Power Estimator
244
Differential Detector
250
atan
254
Frequency Calculator
258

FIG. 18

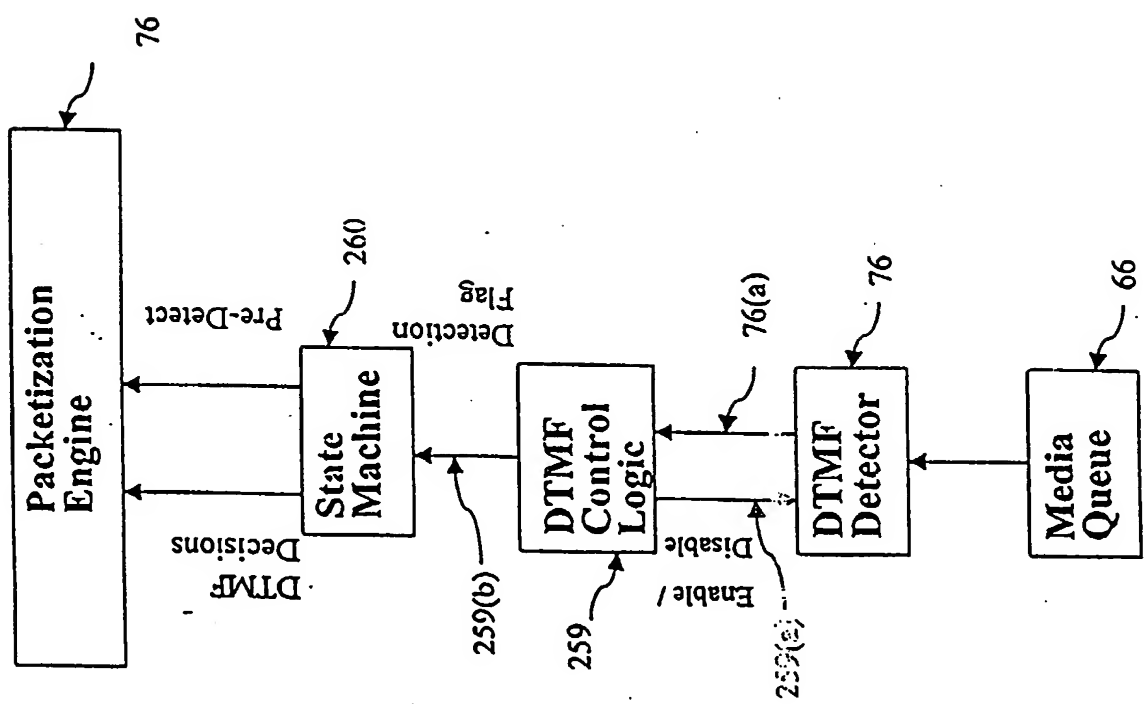

76
Packetization Engine
Pre-Detect
DTMF Decisions
260
State Machine
Detection Flag
259(b)
DTMF Control Logic
259
76(a)
Enable / Disable
76
DTMF Detector
66
Media Queue

FIG. 18A

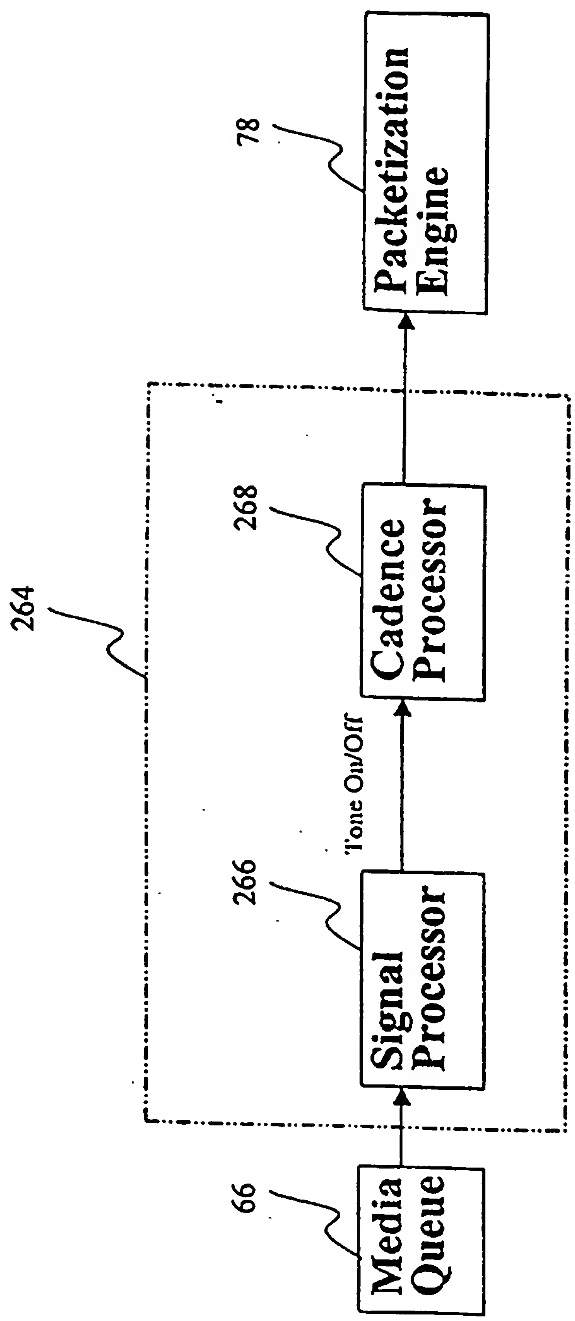

Packetization Engine
78
Cadence Processor
268
264
Tone On/Off
Signal Processor
266
Media Queue
66

FIG. 19

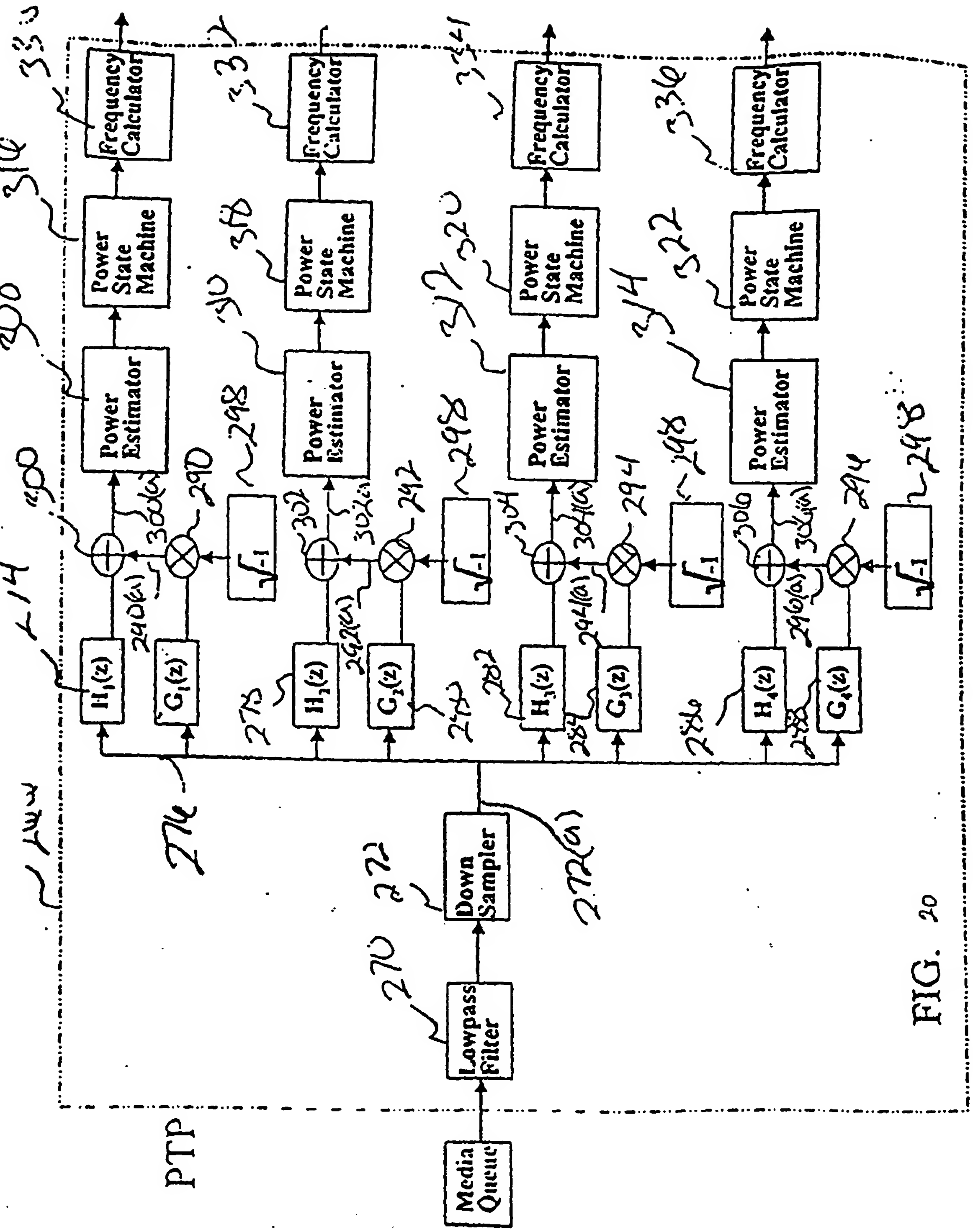

PTP
Media Queue
Lowpass Filter
Down Sampler
$H_1(z)$
$G_1(z)$
$H_2(z)$
$G_2(z)$
$H_3(z)$
$G_3(z)$
$H_4(z)$
$G_4(z)$
$\sqrt{-1}$
Power Estimator
Power State Machine
Frequency Calculator
270
272
272(a)
274
276
278
280
282
284
286
288
290
290(a)
292
292(a)
294
294(a)
296
296(a)
298
300
300(a)
302
302(a)
304
304(a)
306
306(a)
308
310
312
314
316
318
320
322
330
332
334
336

FIG. 20

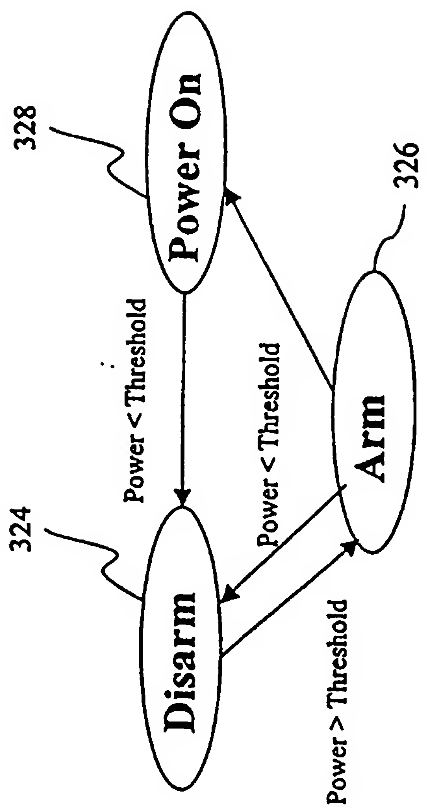

328
Power On
Power < Threshold
326
Power < Threshold
Arm
324
Disarm
Power > Threshold

FIG. 21

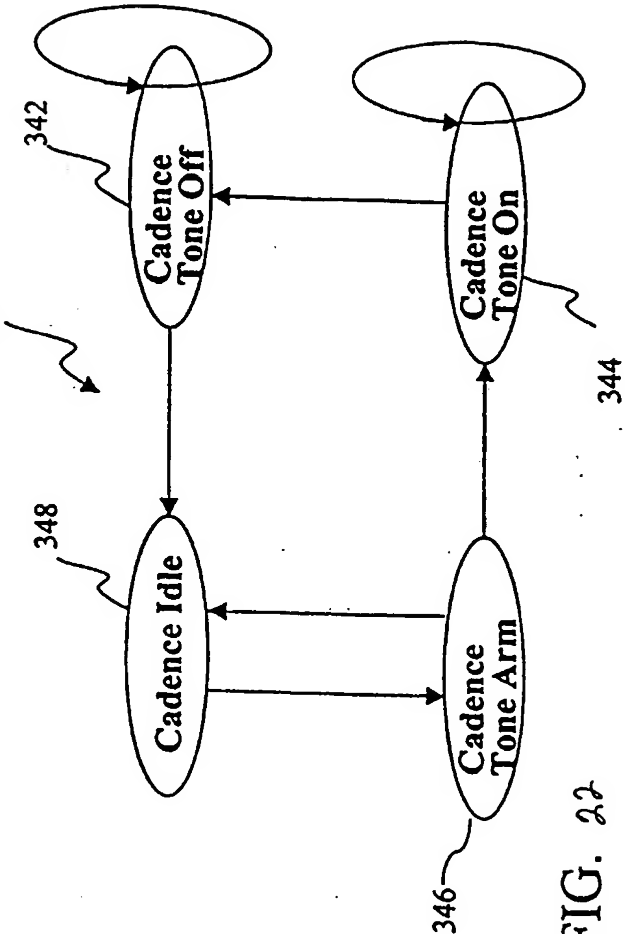

340
342
Cadence Tone Off
Cadence Tone On
344
348
Cadence Idle
Cadence Tone Arm
346

FIG. 22

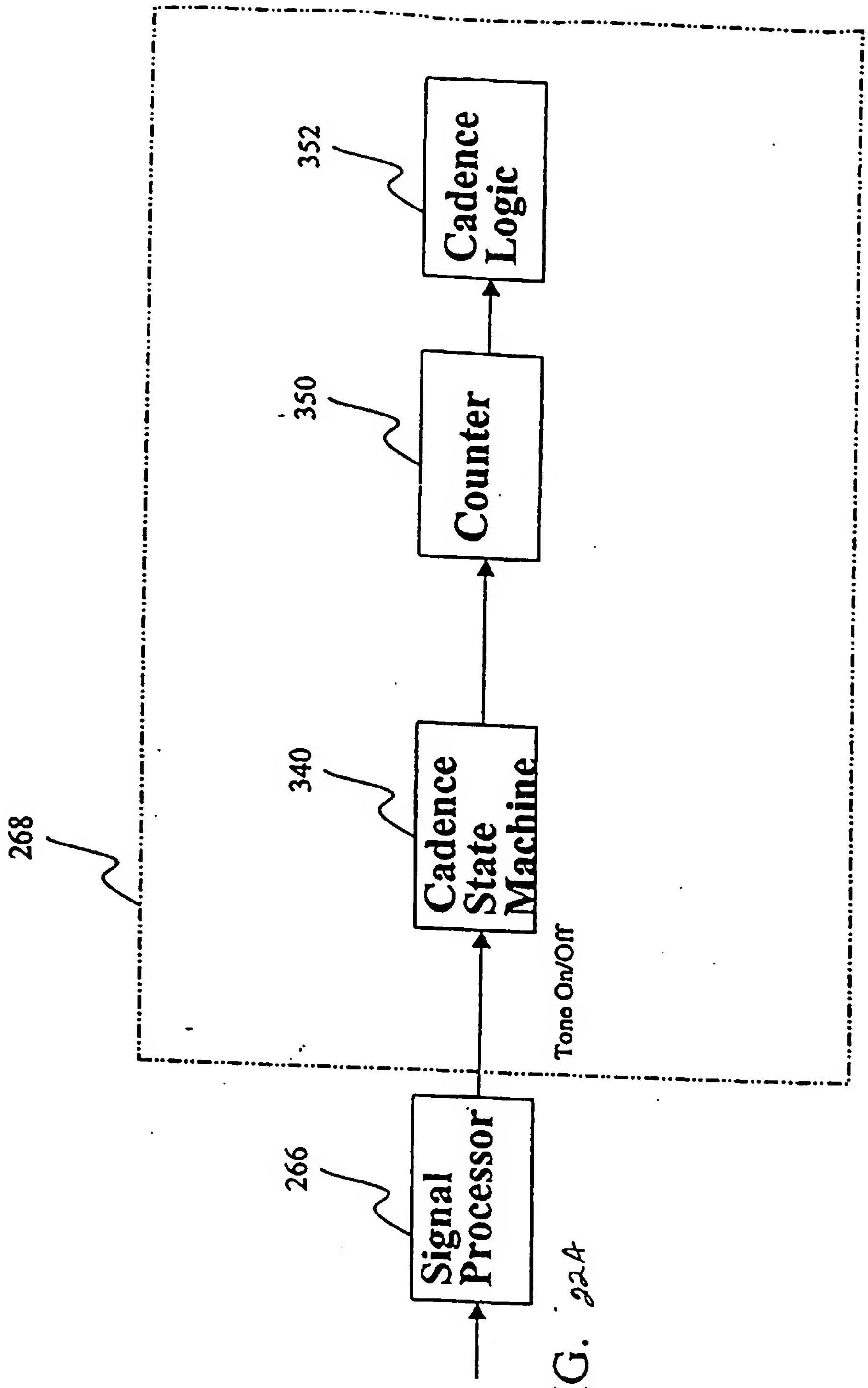

268
266
Signal Processor
Tone On/Off
340
Cadence State Machine
350
Counter
352
Cadence Logic

FIG. 22A

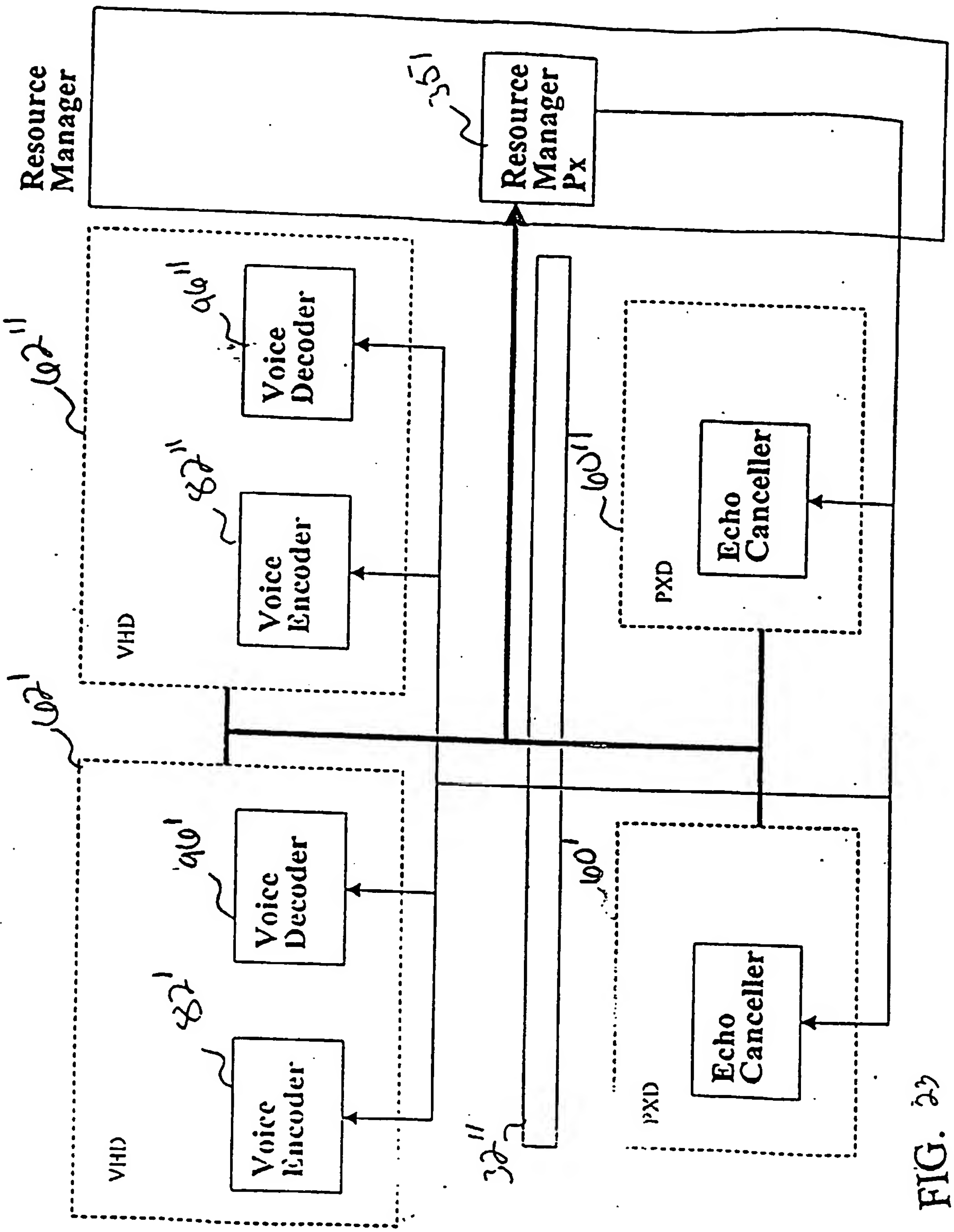

Resource Manager
Resource Manager Px
VHD
Voice Encoder
Voice Decoder
VHD
Voice Encoder
Voice Decoder
PXD
Echo Canceller
PXD
Echo Canceller

FIG. 23

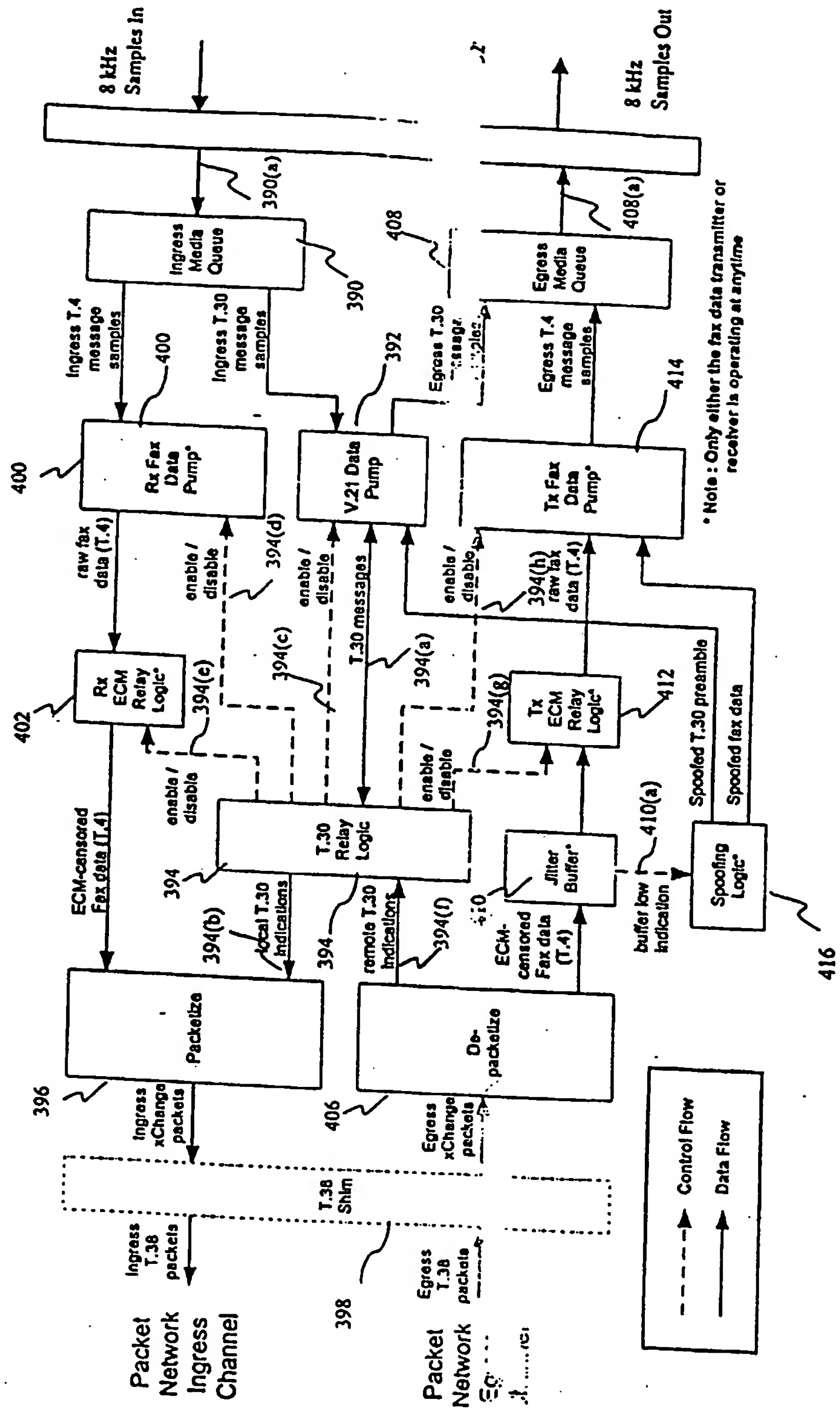

8 kHz Samples In
8 kHz Samples Out
390(a)
408(a)
Ingress Media Queue
390
408
Egress Media Queue
Ingress T.4 message samples
Ingress T.30 message samples
Egress T.4 message samples
400
392
414
Rx Fax Data Pump*
V.21 Data Pump
Tx Fax Data Pump*
* Note : Only either the fax data transmitter or receiver is operating at anytime
raw fax data (T.4)
enable / disable
394(d)
T.30 messages
394(h)
394(c)
394(a)
402
Rx ECM Relay Logic*
394(e)
Tx ECM Relay Logic*
412
394(g)
Spoofed T.30 preamble
Spoofed fax data
ECM-censored Fax data (T.4)
T.30 Relay Logic
394
Jitter Buffer*
410(a)
buffer low indication
Spoofing Logic*
416
394(b)
local T.30 indications
remote T.30 indications
394(f)
Packetize
De-packetize
396
406
Ingress xChange packets
Egress xChange packets
T.38 Shim
398
Ingress T.38 packets
Egress T.38 packets
Packet Network Ingress Channel
Packet Network
Control Flow
Data Flow

FIG. 24

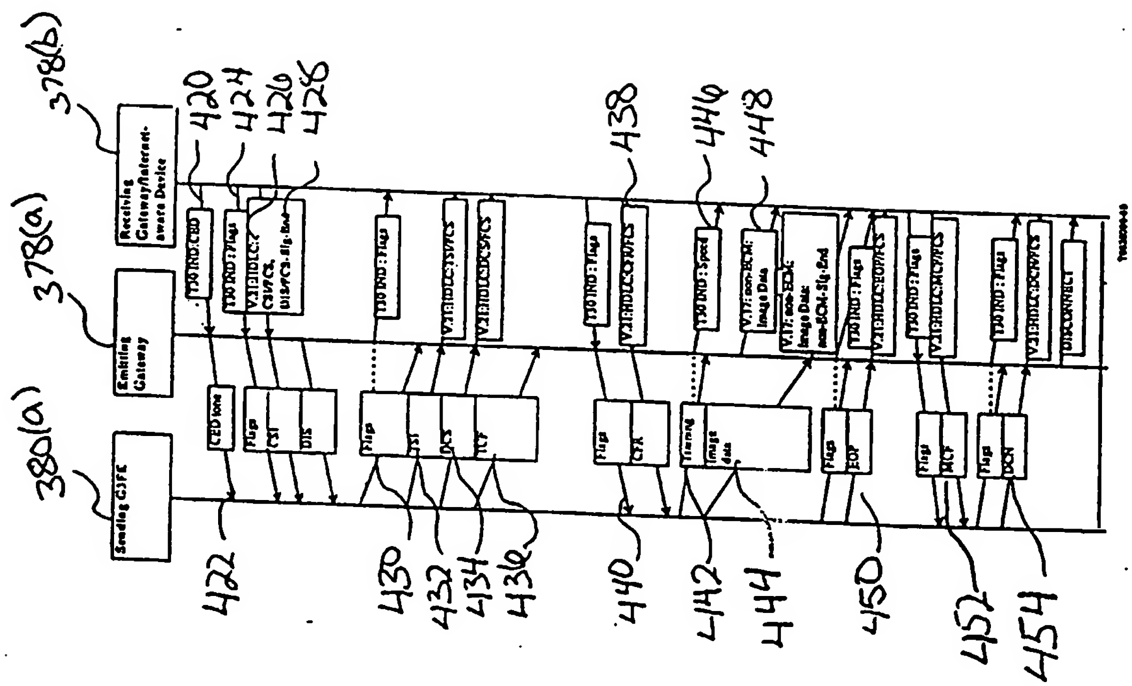

380(a)
378(a)
378(b)
Sending G3FE
Emitting Gateway
Receiving Gateway/Internet-aware Device
420
422
424
426
428
430
432
434
436
438
440
442
444
446
448
450
452
454

FIG. 25

Fax Spoofing

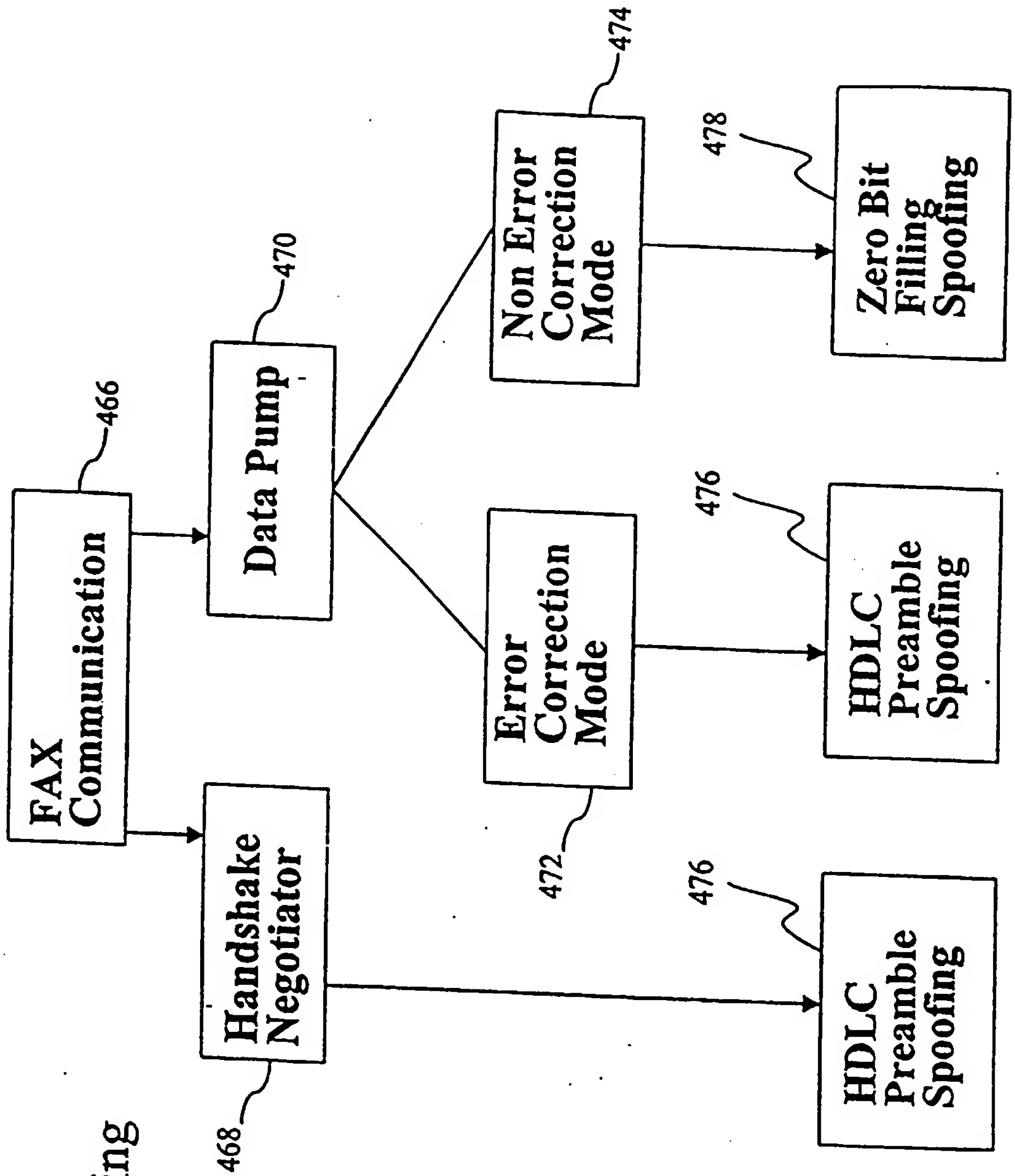


FIG. 26

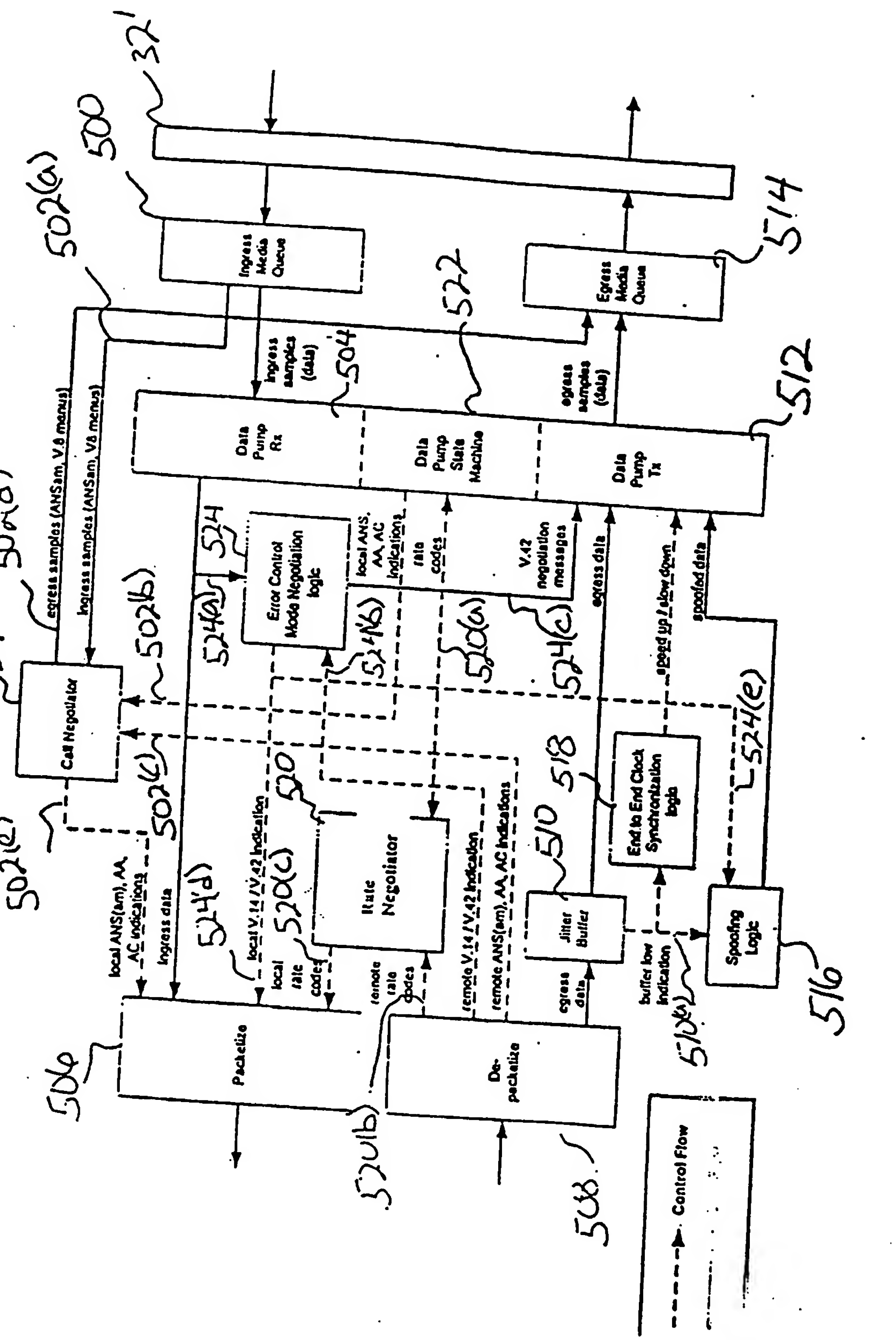

32'
500
502(a)
Ingress Media Queue
Egress Media Queue
514
522
504
512
egress samples (ANSam, V.8 menus)
ingress samples (ANSam, V8 menus)
ingress samples (data)
egress samples (data)
Data Pump Rx
Data Pump State Machine
Data Pump Tx
504
Call Negotiator
502(b)
524
Error Control Mode Negotiation logic
local ANS, AA, AC indications
rate codes
V.42 negotiation messages
egress data
speed up / slow down
spoofed data
524(a)
524(b)
520(a)
524(c)
524(e)
502(c)
502(e)
local ANS(am), AA, AC indications
ingress data
524(d)
local V.14 / V.42 indication
520(c)
local rate codes
520
Rate Negotiator
remote rate codes
remote V.14 / V.42 indication
remote ANS(am), AA, AC indications
510
Jitter Buffer
518
End to End Clock Synchronization logic
Spoofing Logic
516
buffer low indication
510(a)
Packetize
De-packetize
506
520(b)
508
Control Flow

FIG. 27

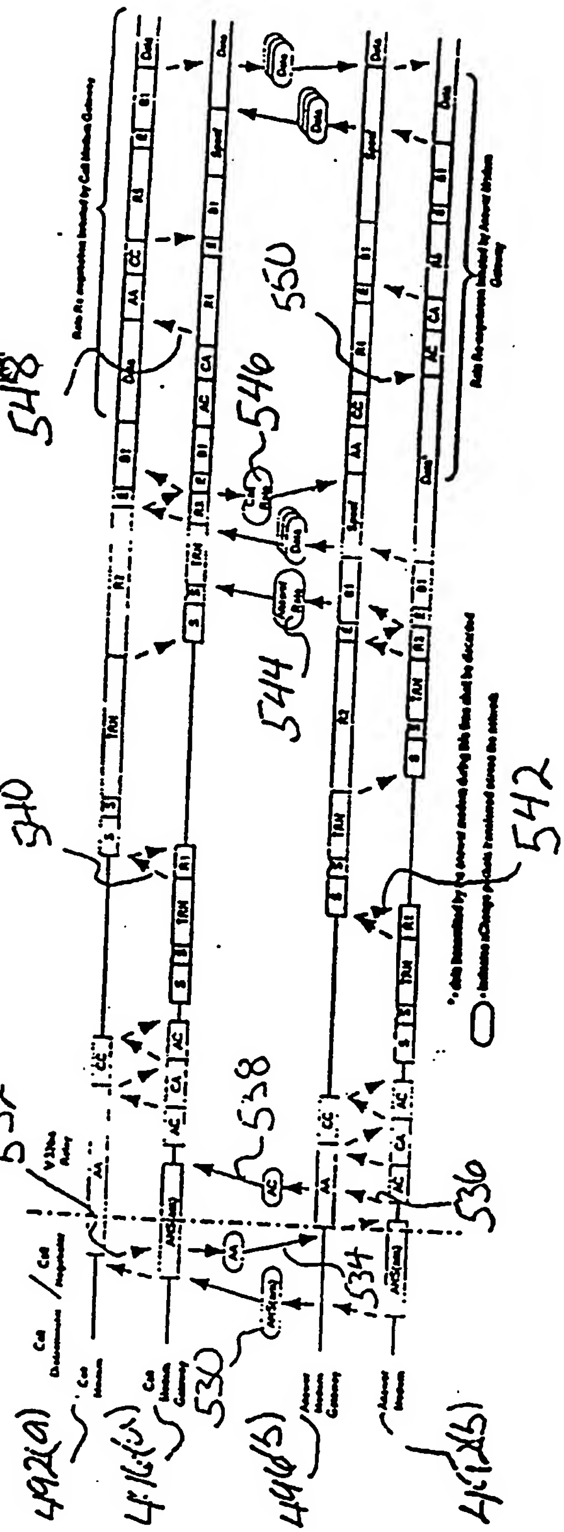


FIG. 28

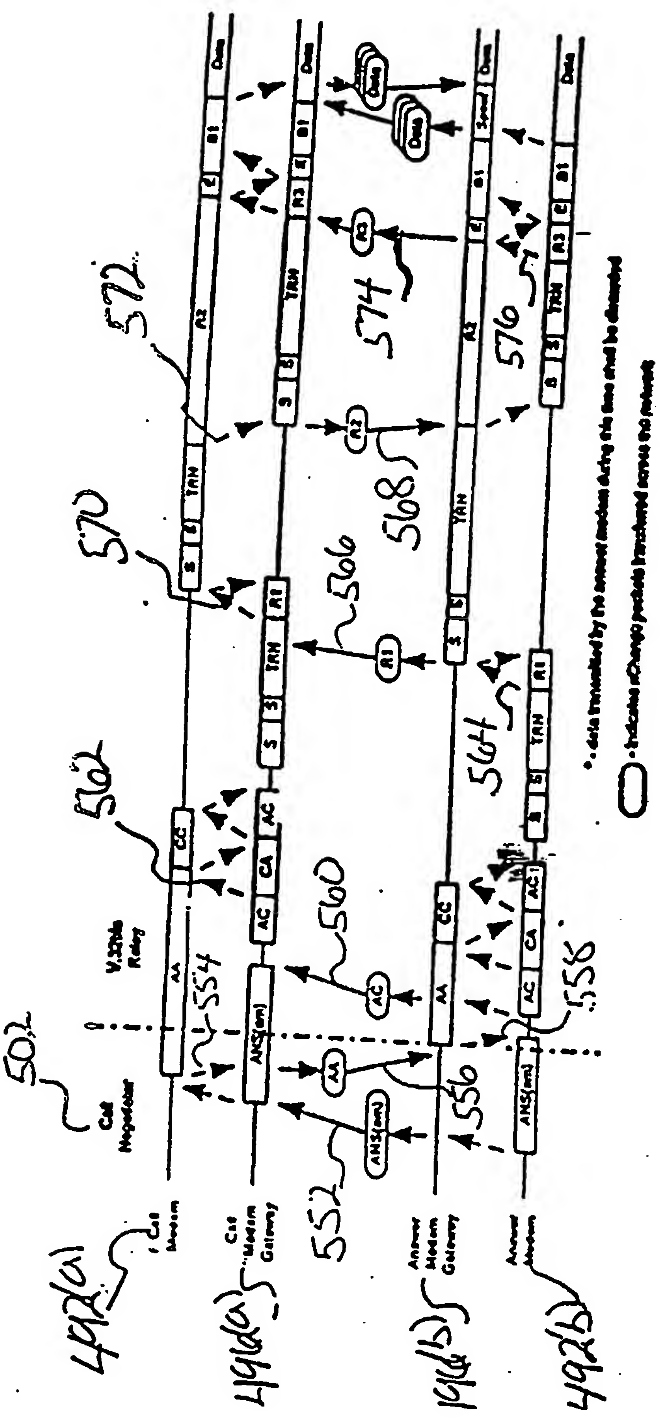

502
554
562
570
572
560
566
568
574
576
552
556
558
564
492(a)
496(a)
496(b)
492(b)
AA
CC
AC
CA
TRN
R1
R2
R3
B1
Data

FIG. 29

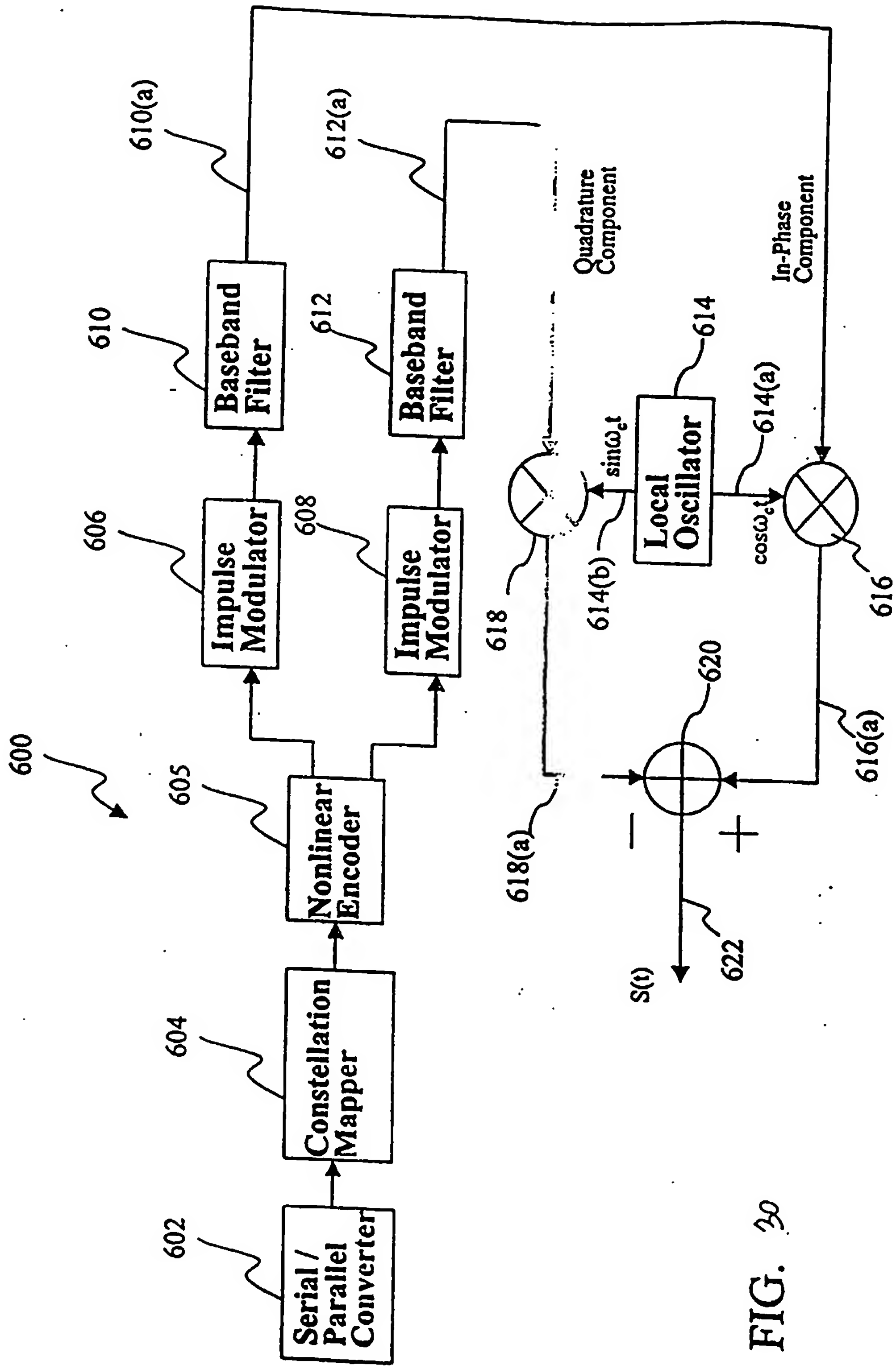

600
602
Serial / Parallel Converter
604
Constellation Mapper
605
Nonlinear Encoder
606
Impulse Modulator
608
Impulse Modulator
610
Baseband Filter
612
Baseband Filter
610(a)
612(a)
Quadrature Component
In-Phase Component
614
Local Oscillator
614(a)
614(b)
$\sin\omega_c t$
$\cos\omega_c t$
616
616(a)
618
618(a)
620
−
+
622
S(t)

FIG. 30

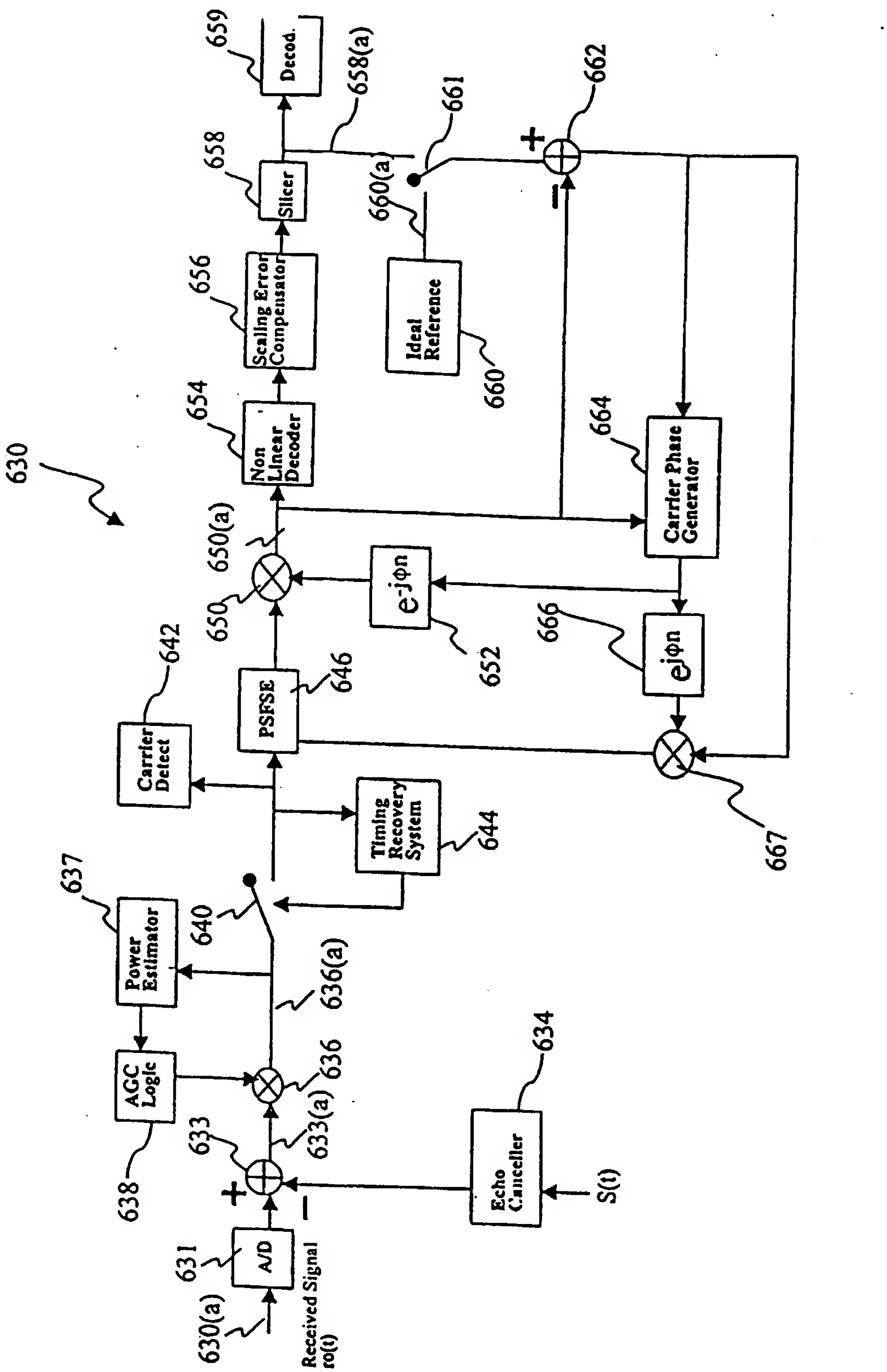

630
Decod.
659
658(a)
Slicer
658
661
662
+
−
660(a)
Scaling Error Compensator
656
Ideal Reference
660
Non Linear Decoder
654
664
Carrier Phase Generator
650(a)
650
e-jφn
652
666
ejφn
642
Carrier Detect
PSFSE
646
667
Timing Recovery System
644
637
Power Estimator
640
636(a)
AGC Logic
636
638
633
633(a)
634
Echo Canceller
S(t)
+
−
A/D
631
630(a)
Received Signal
ro(t)

FIG. 31

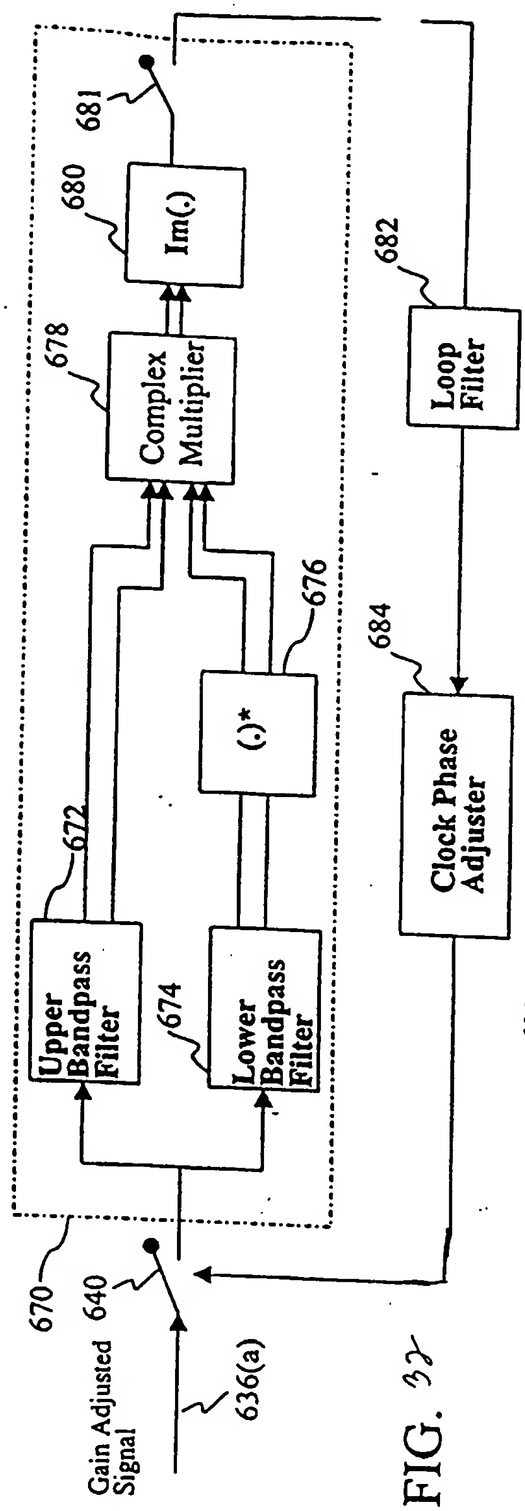

Gain Adjusted Signal
636(a)
640
670
Upper Bandpass Filter
672
Lower Bandpass Filter
674
(.)*
676
Complex Multiplier
678
Im(.)
680
681
Loop Filter
682
Clock Phase Adjuster
684

FIG. 32

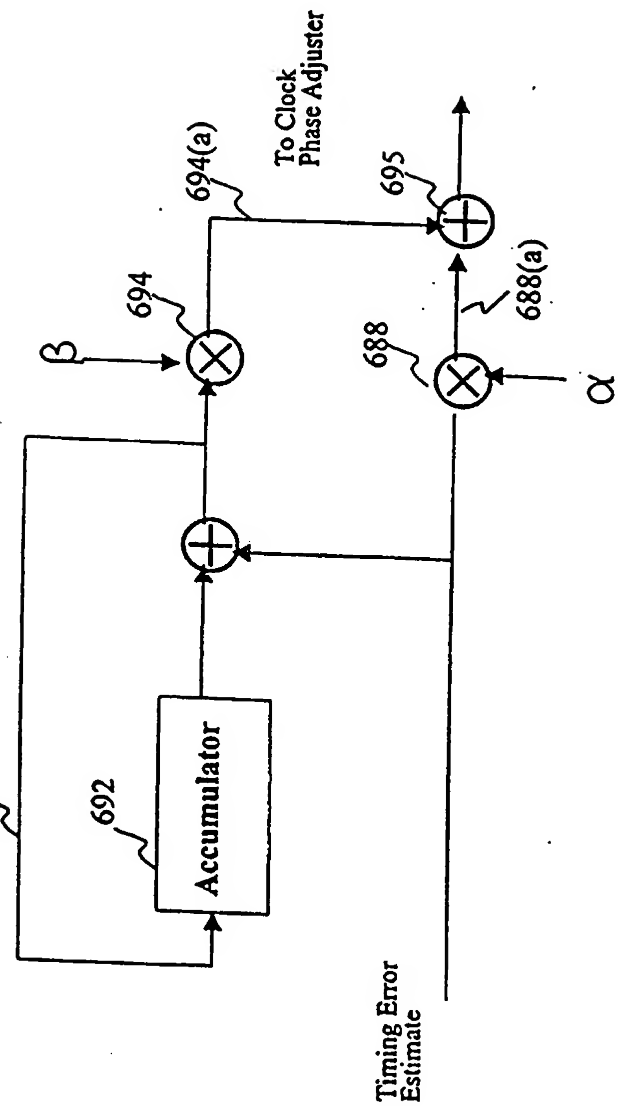

690
Accumulator
692
β
694
694(a)
To Clock Phase Adjuster
695
688
688(a)
α
Timing Error Estimate

FIG. 33

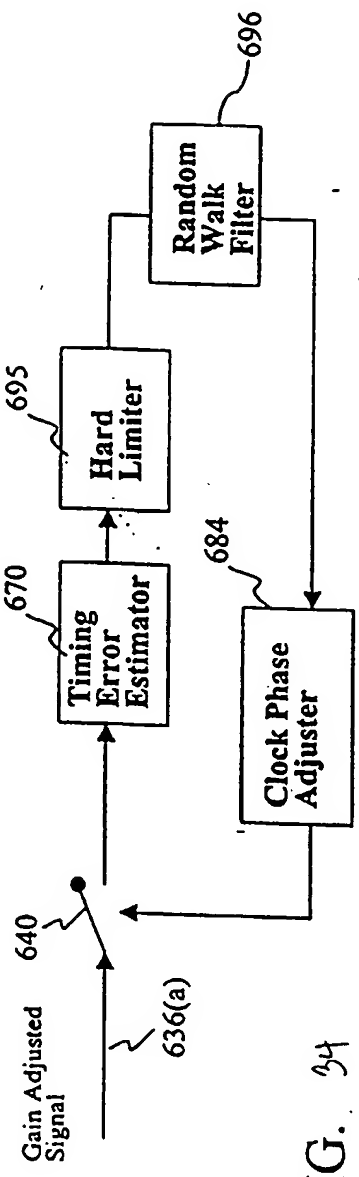

Gain Adjusted Signal
636(a)
640
670
Timing Error Estimator
695
Hard Limiter
696
Random Walk Filter
684
Clock Phase Adjuster

FIG. 34

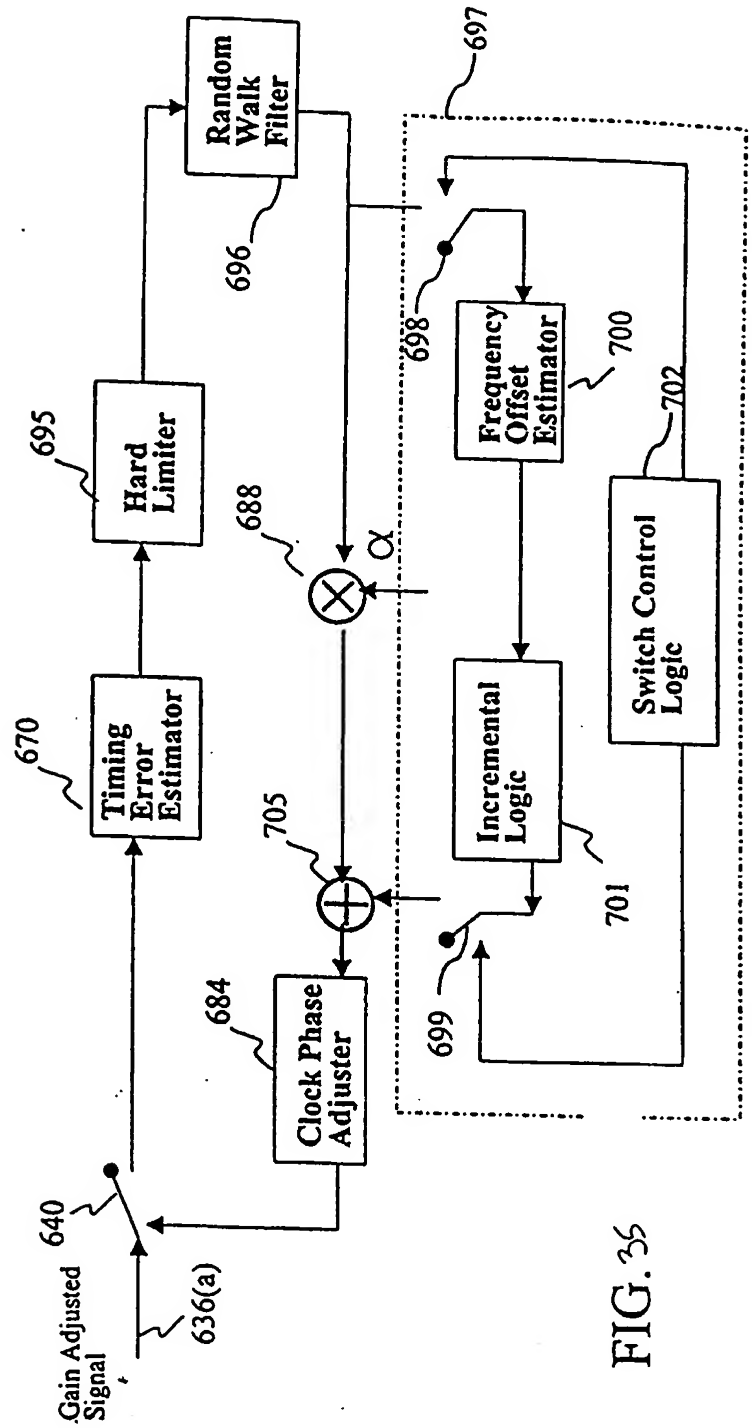

Gain Adjusted Signal
636(a)
640
684
Clock Phase Adjuster
670
Timing Error Estimator
695
Hard Limiter
696
Random Walk Filter
688
α
705
697
698
Frequency Offset Estimator
700
702
Switch Control Logic
Incremental Logic
701
699

FIG. 35

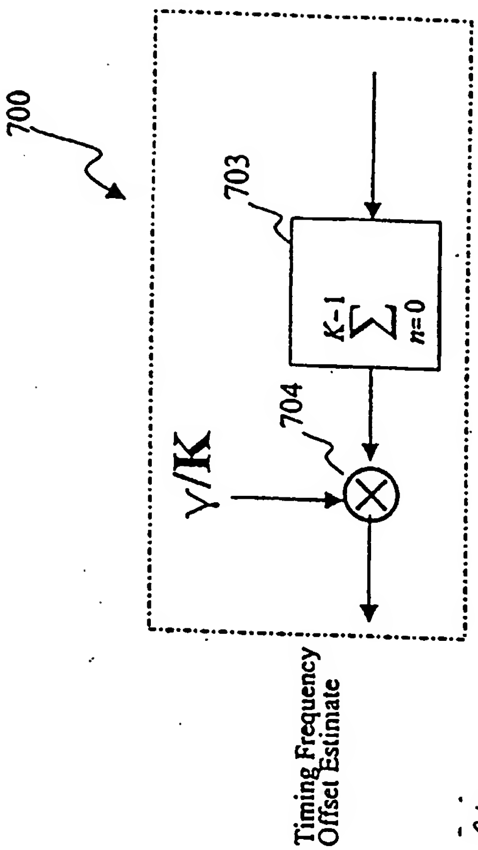


FIG. 36

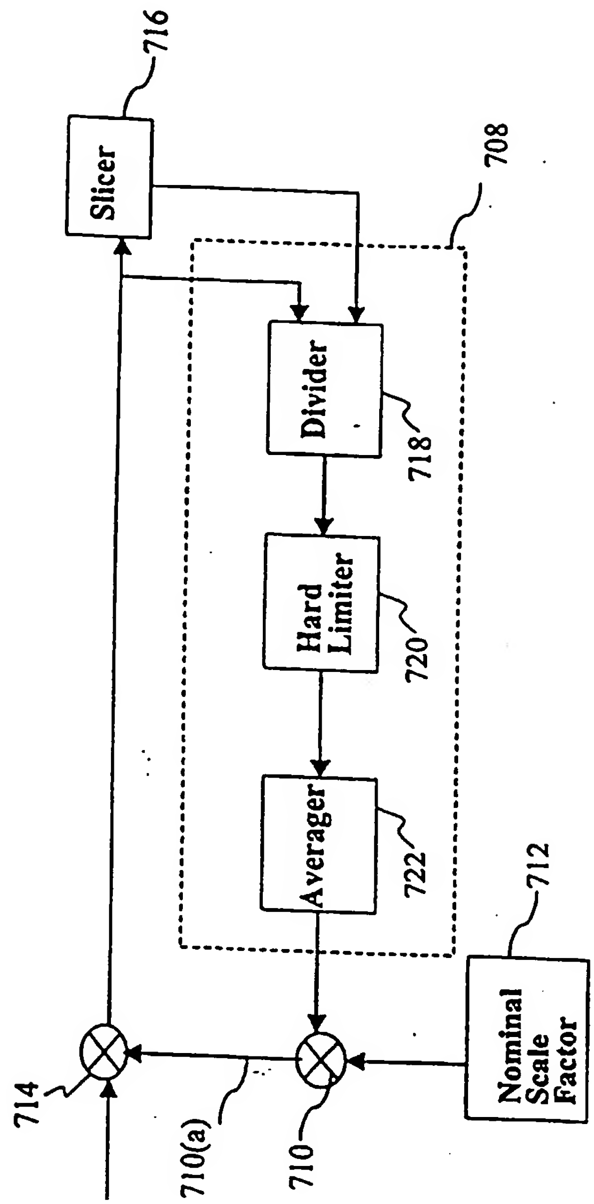

Slicer
716
708
Divider
718
Hard Limiter
720
Averager
722
712
Nominal Scale Factor
714
710(a)
710

FIG. 37

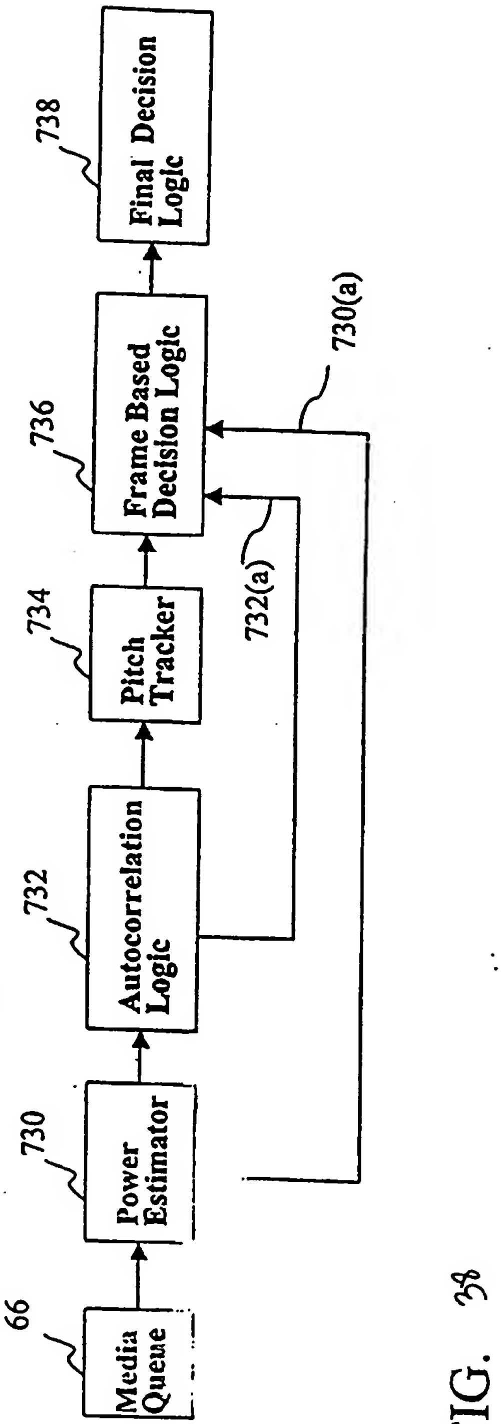

Final Decision Logic
738
Frame Based Decision Logic
736
730(a)
Pitch Tracker
734
732(a)
Autocorrelation Logic
732
Power Estimator
730
Media Queue
66

FIG. 38

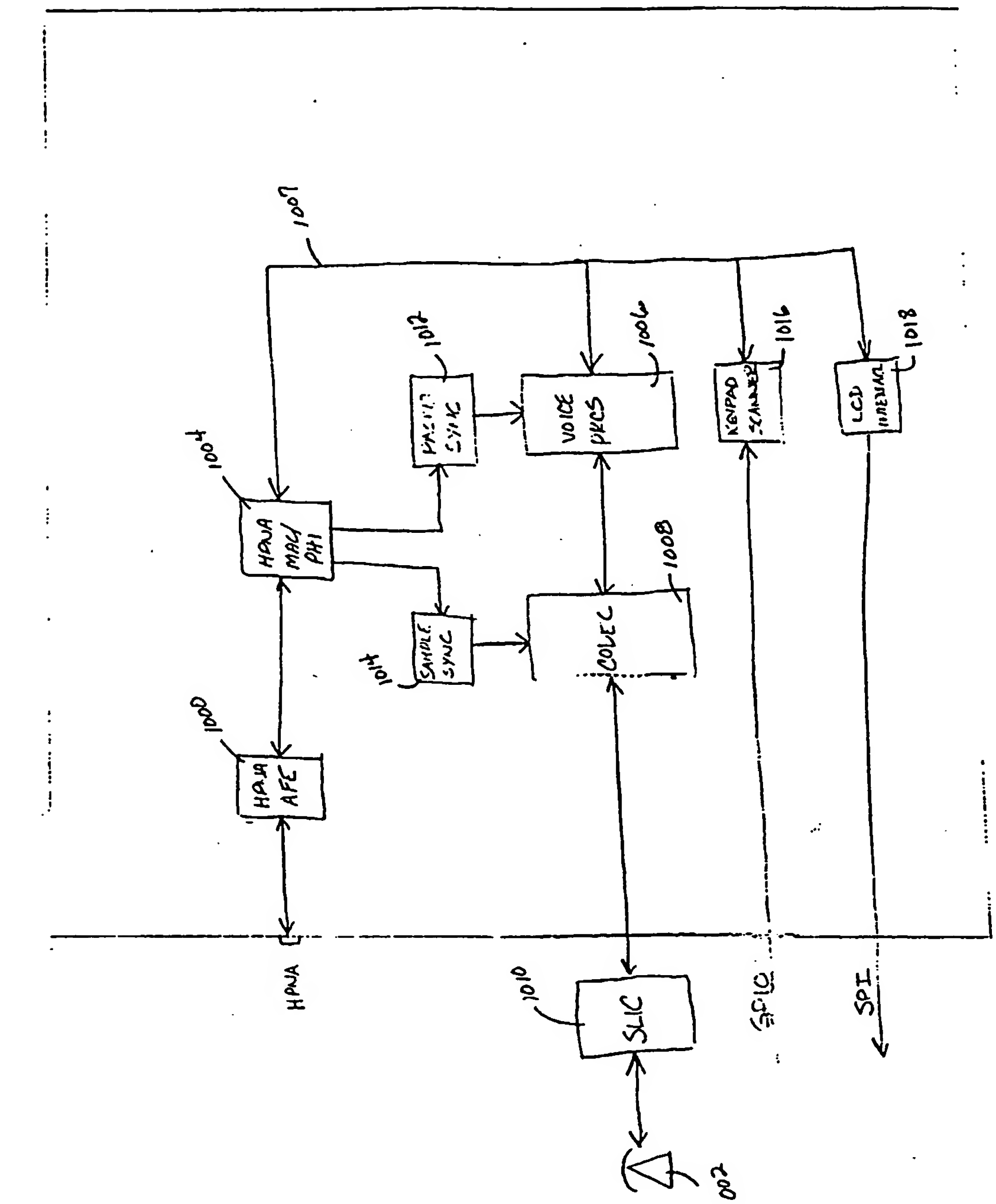

1007
1004
HPNA MAC/PHY
1012
PACKET SYNC
VOICE PKCS
1006
KEYPAD SCANNER
1016
LCD INTERFACE
1018
1000
HPNA AFE
1008
CODEC
SAMPLE SYNC
1014
HPNA
1010
SLIC
GPIO
SPI
1002

FIG. 39

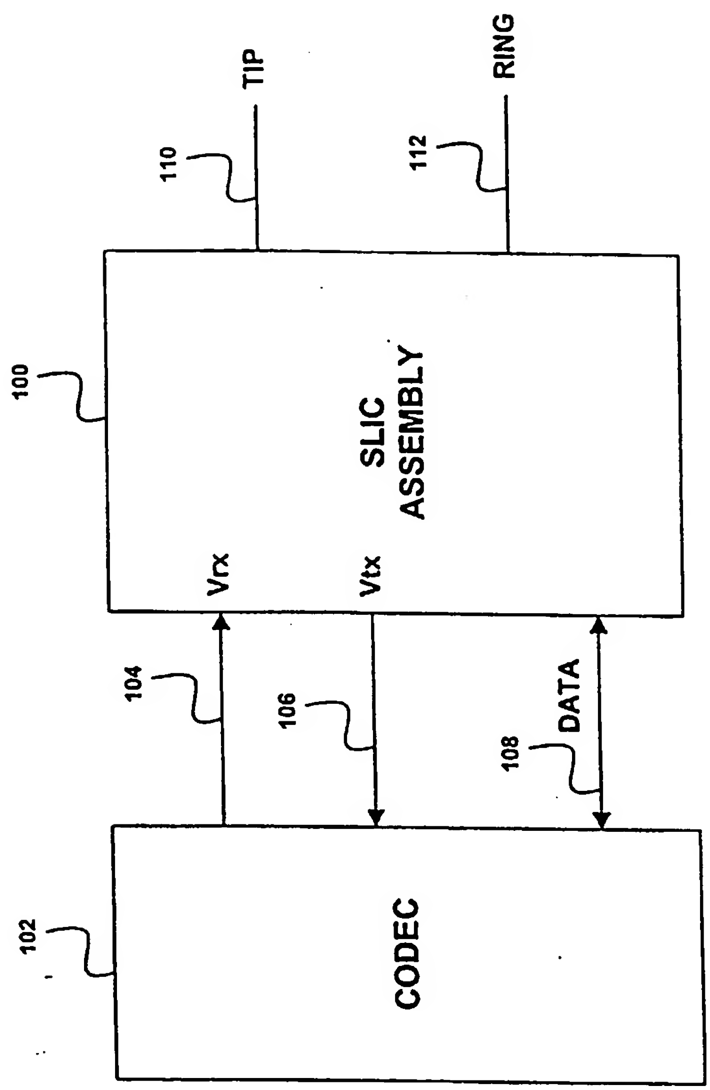

TIP
110
RING
112
100
SLIC
ASSEMBLY
Vrx
Vtx
104
106
108
DATA
102
CODEC

FIG. 40

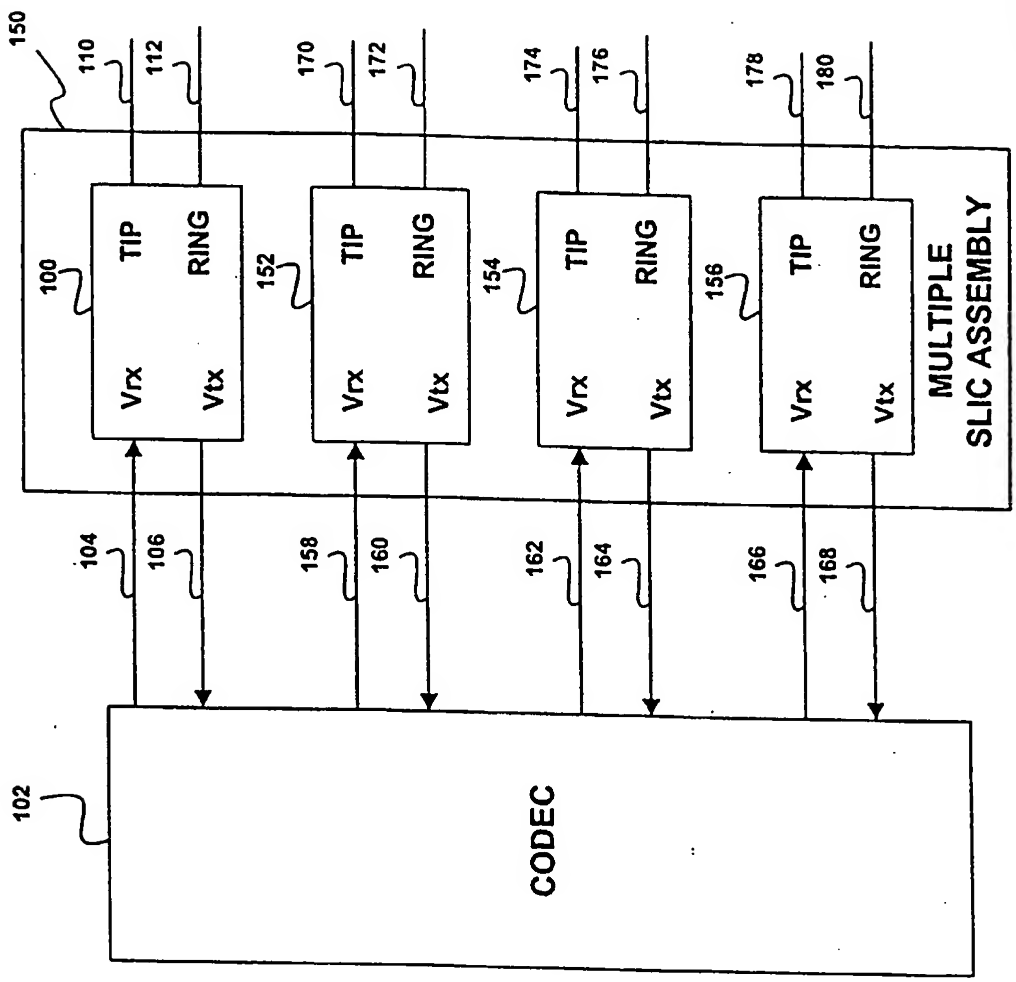

150
110
112
170
172
174
176
178
180
100
TIP
RING
Vrx
Vtx
152
TIP
RING
Vrx
Vtx
154
TIP
RING
Vrx
Vtx
156
TIP
RING
Vrx
Vtx
MULTIPLE
SLIC ASSEMBLY
104
106
158
160
162
164
166
168
102
CODEC

FIG. 41

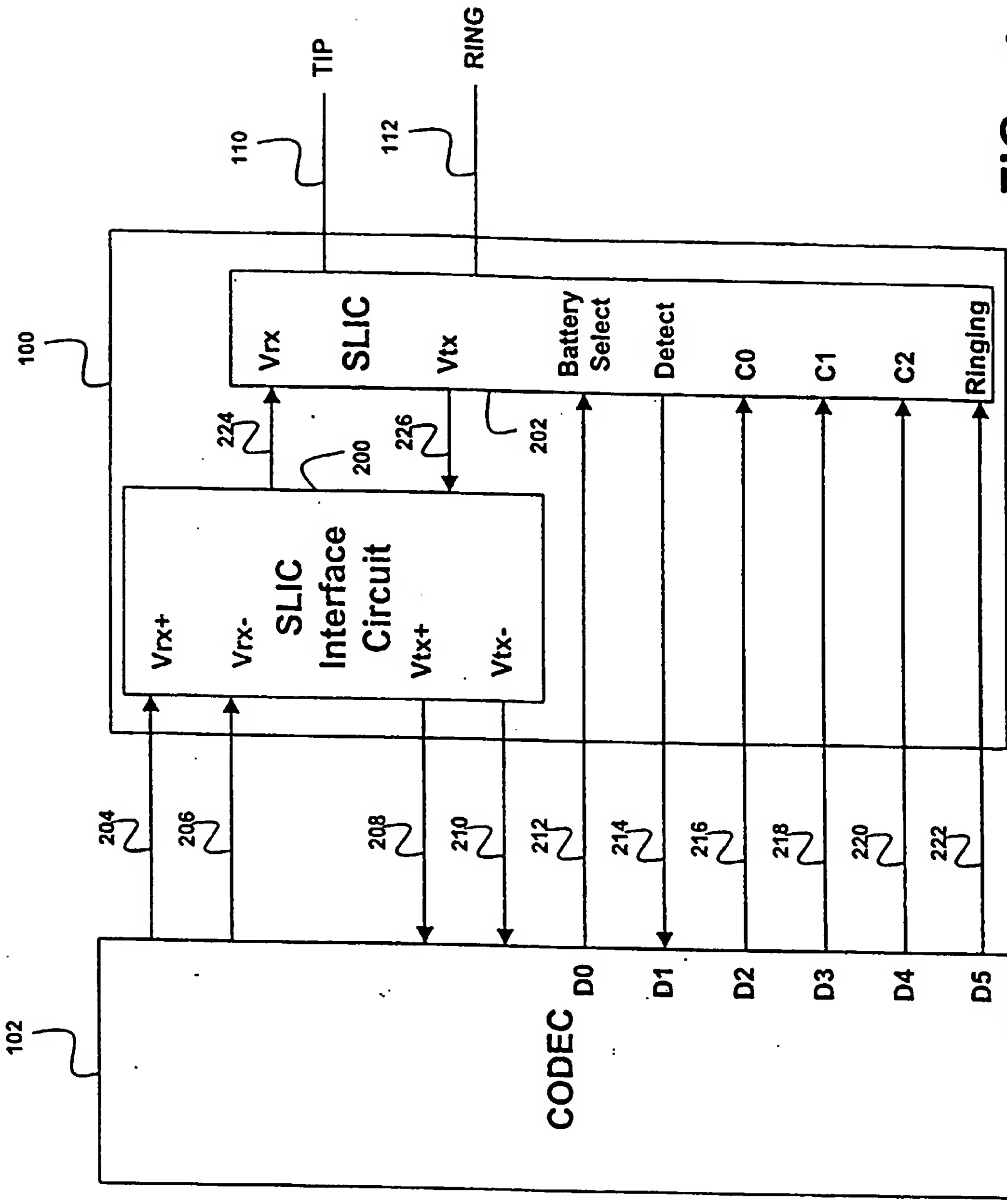

TIP
RING
110
112
100
SLIC
Vrx
Vtx
Battery Select
Detect
C0
C1
C2
Ringing
202
224
226
200
SLIC Interface Circuit
Vrx+
Vrx-
Vtx+
Vtx-
204
206
208
210
212
214
216
218
220
222
CODEC
D0
D1
D2
D3
D4
D5
102

FIG. 42

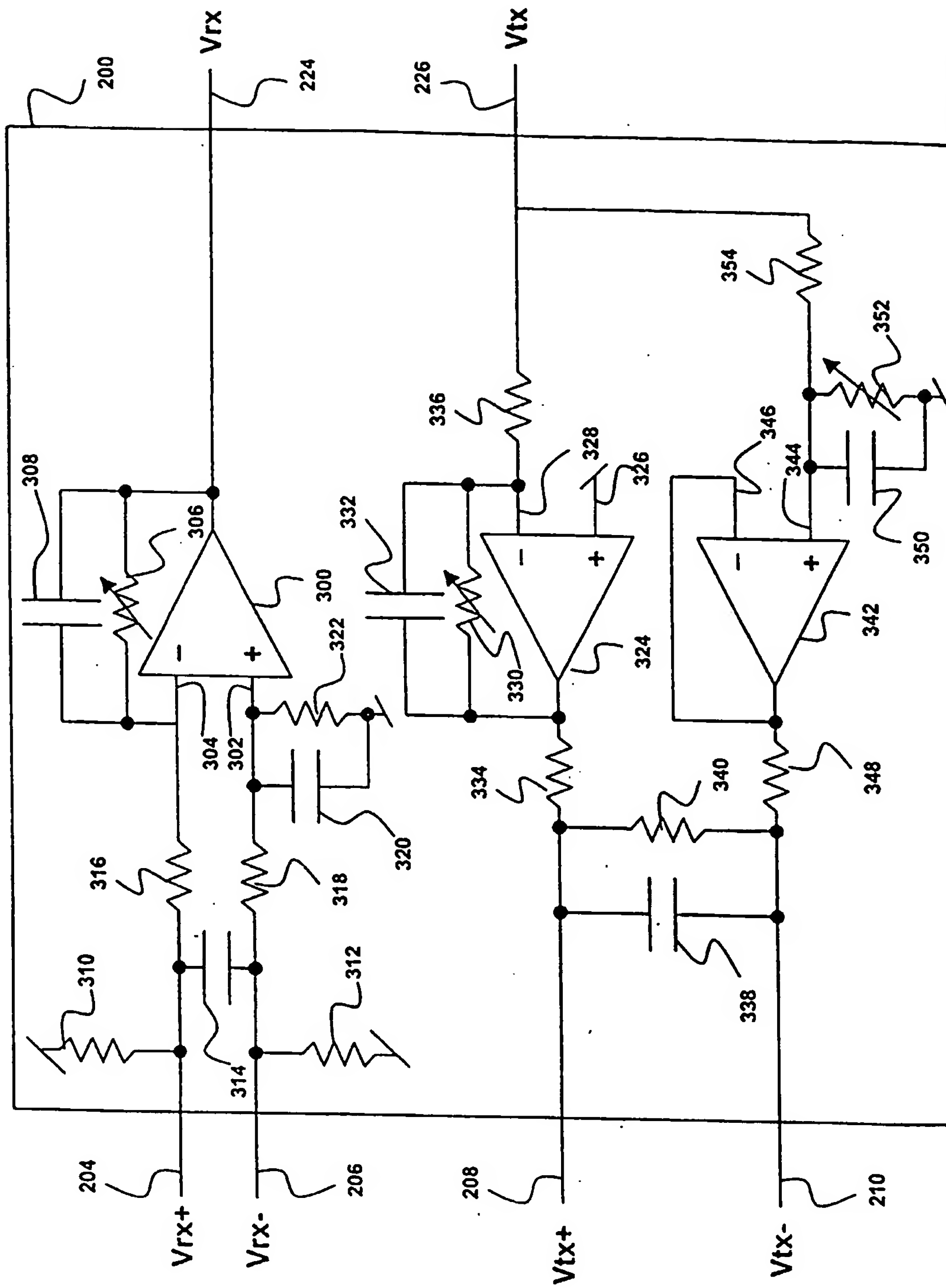

200
Vrx
224
226
Vtx
354
352
336
346
344
308
306
328
326
332
300
350
322
342
324
330
304
302
334
340
348
320
316
318
310
312
338
314
204
206
208
210
Vrx+
Vrx-
Vtx+
Vtx-

FIG. 43

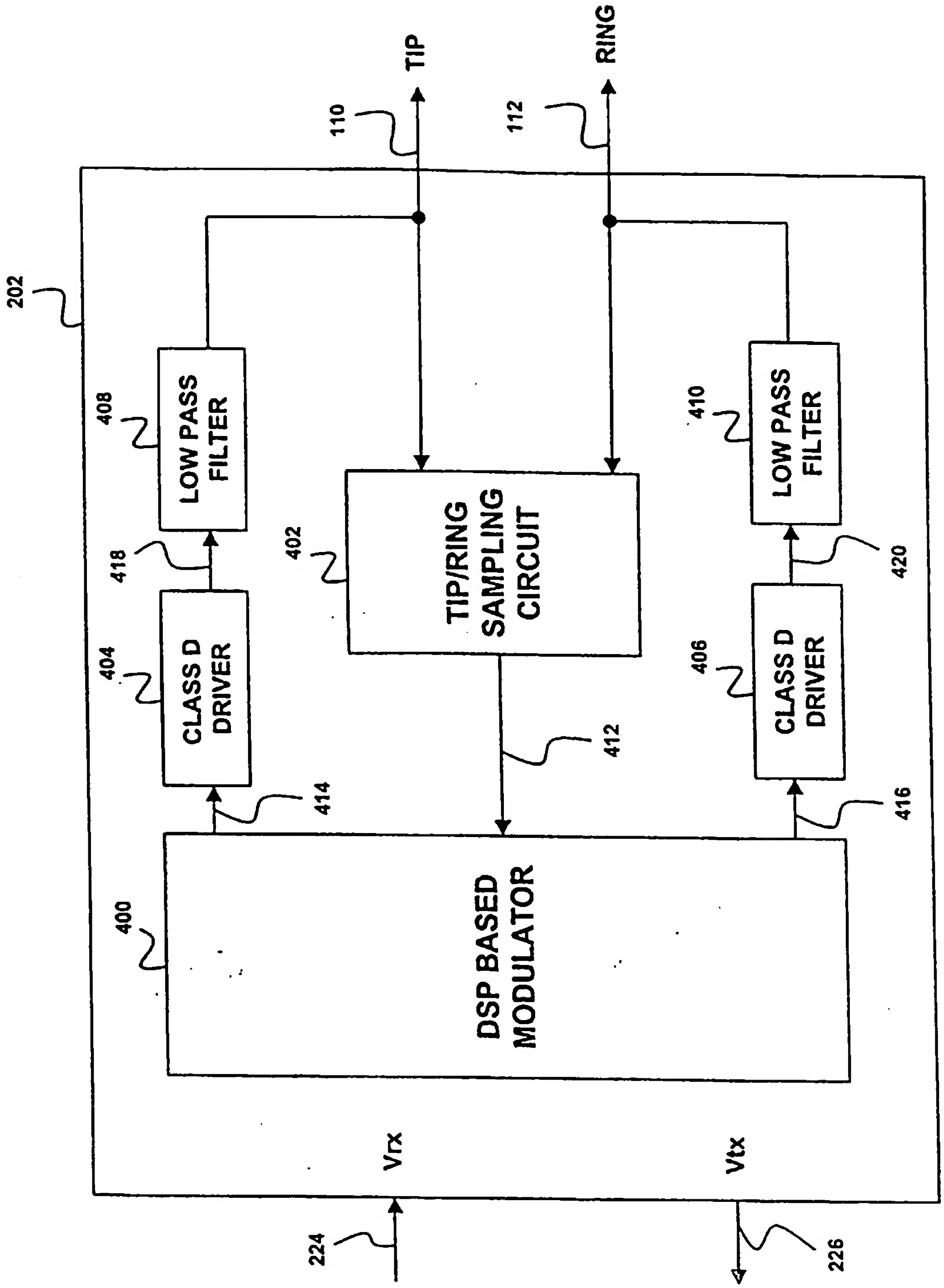

TIP
RING
110
112
202
LOW PASS FILTER
408
LOW PASS FILTER
410
TIP/RING SAMPLING CIRCUIT
402
418
420
CLASS D DRIVER
404
CLASS D DRIVER
406
412
414
416
DSP BASED MODULATOR
400
Vrx
Vtx
224
226

FIG. 44

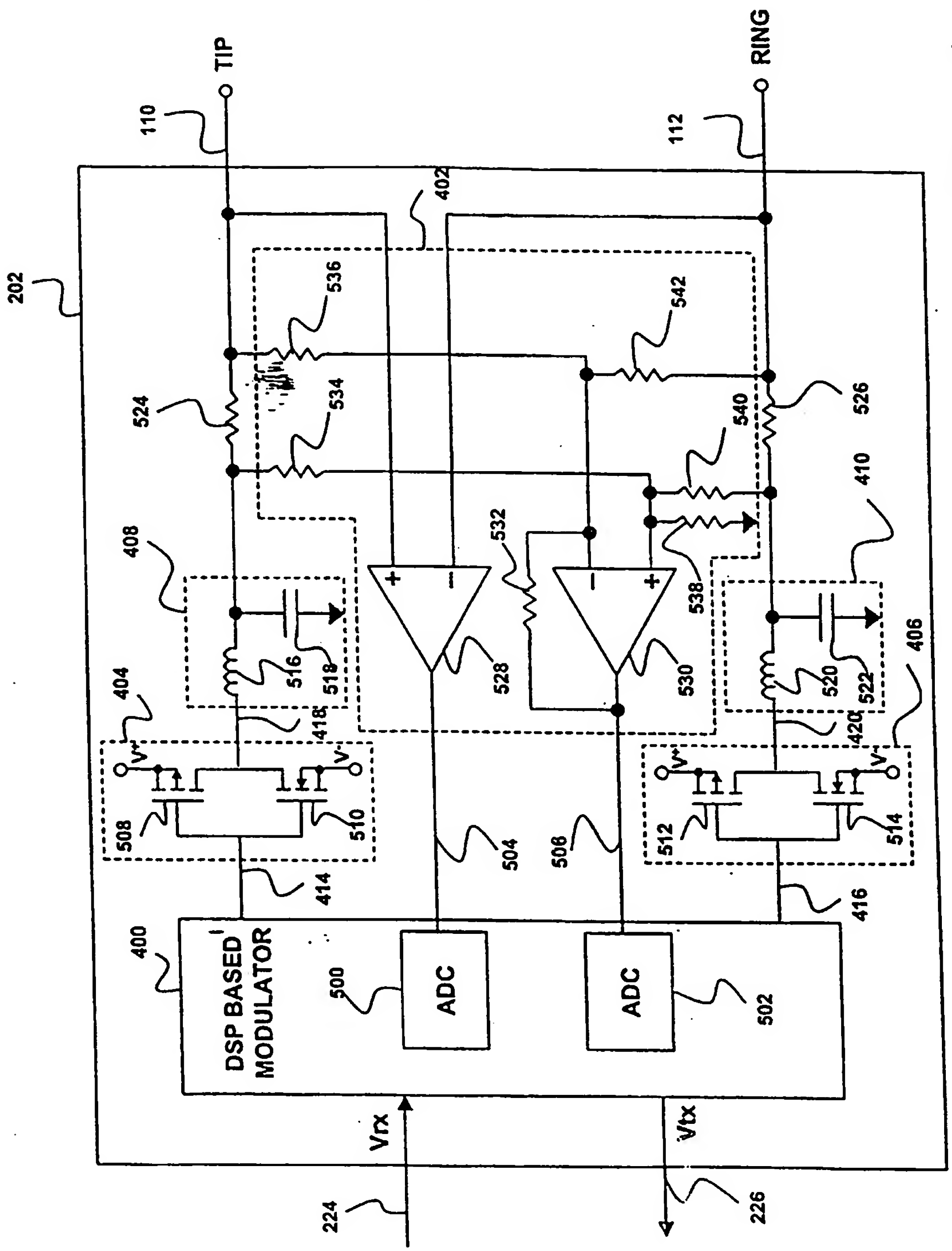

TIP
RING
110
112
202
402
536
542
524
534
526
540
410
408
532
538
404
516
518
528
530
520
522
406
418
420
V+
V-
508
510
504
506
512
514
414
416
400
DSP BASED MODULATOR
500
ADC
ADC
502
Vrx
Vtx
224
226

FIG. 45

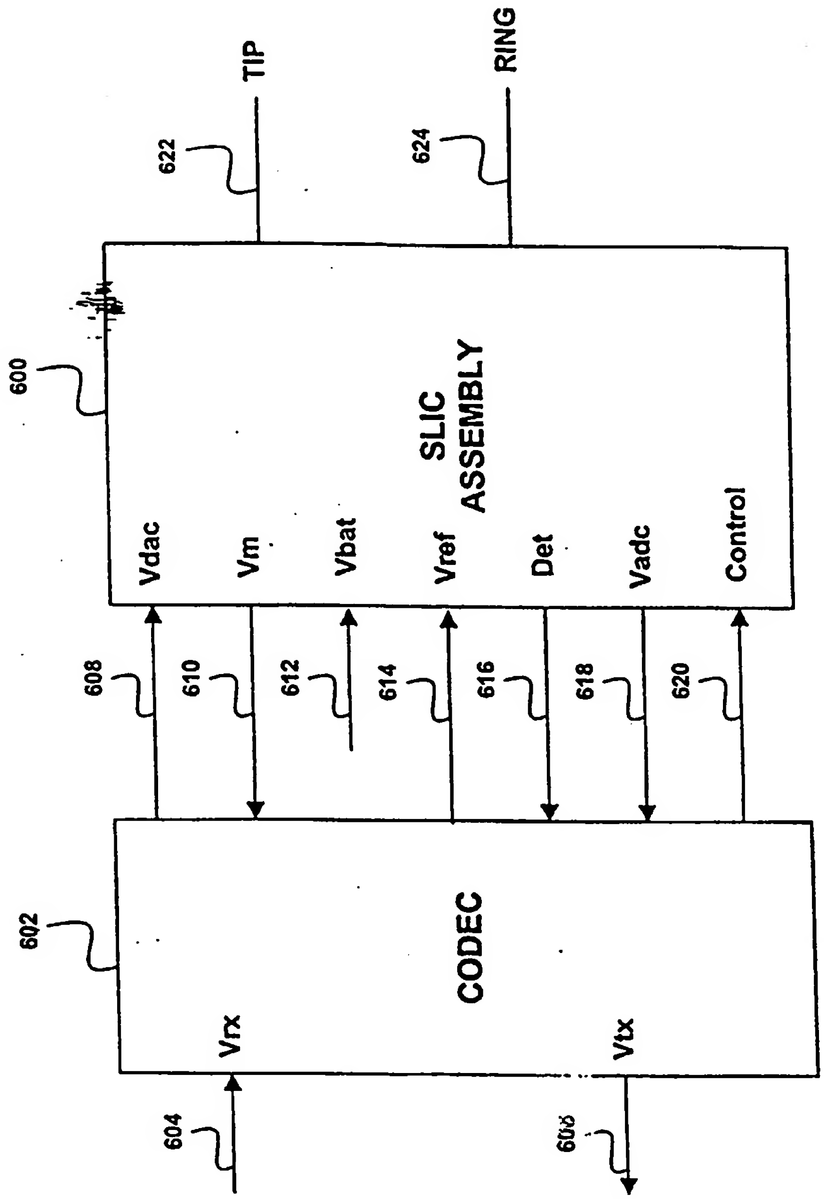
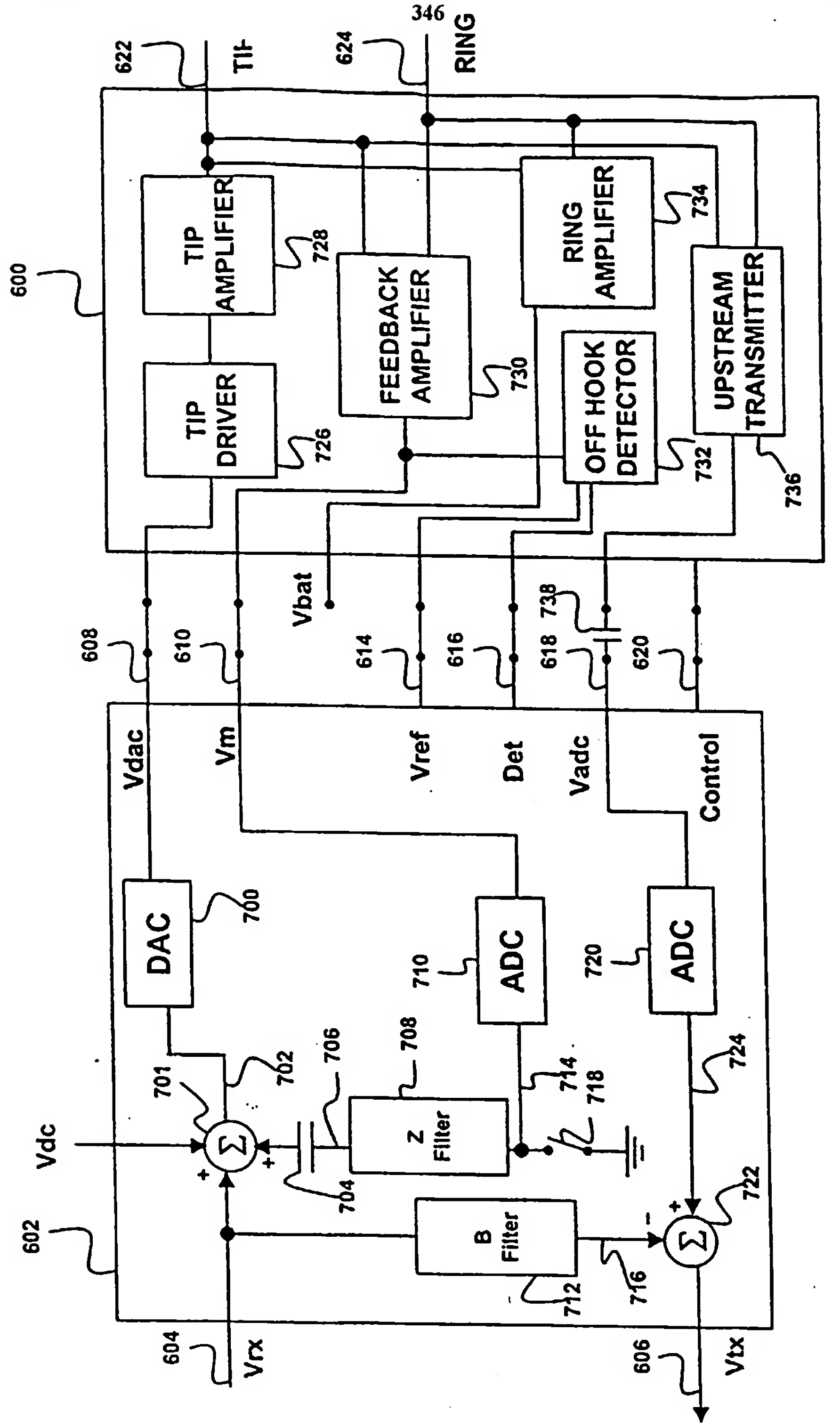

TIP
RING
622
624
600
SLIC
ASSEMBLY
Vdac
Vm
Vbat
Vref
Det
Vadc
Control
608
610
612
614
616
618
620
602
CODEC
Vrx
Vtx
604
606

FIG. 46



622
TIP
624
346
RING
600
TIP AMPLIFIER
728
FEEDBACK AMPLIFIER
730
RING AMPLIFIER
734
TIP DRIVER
726
OFF HOOK DETECTOR
732
UPSTREAM TRANSMITTER
736
608
610
Vbat
614
616
618
738
620
Vdac
Vm
Vref
Det
Vadc
Control
DAC
700
ADC
710
ADC
720
701
702
706
708
714
718
724
Vdc
Z Filter
704
B Filter
712
716
722
602
604
Vrx
606
Vtx

FIG. 47

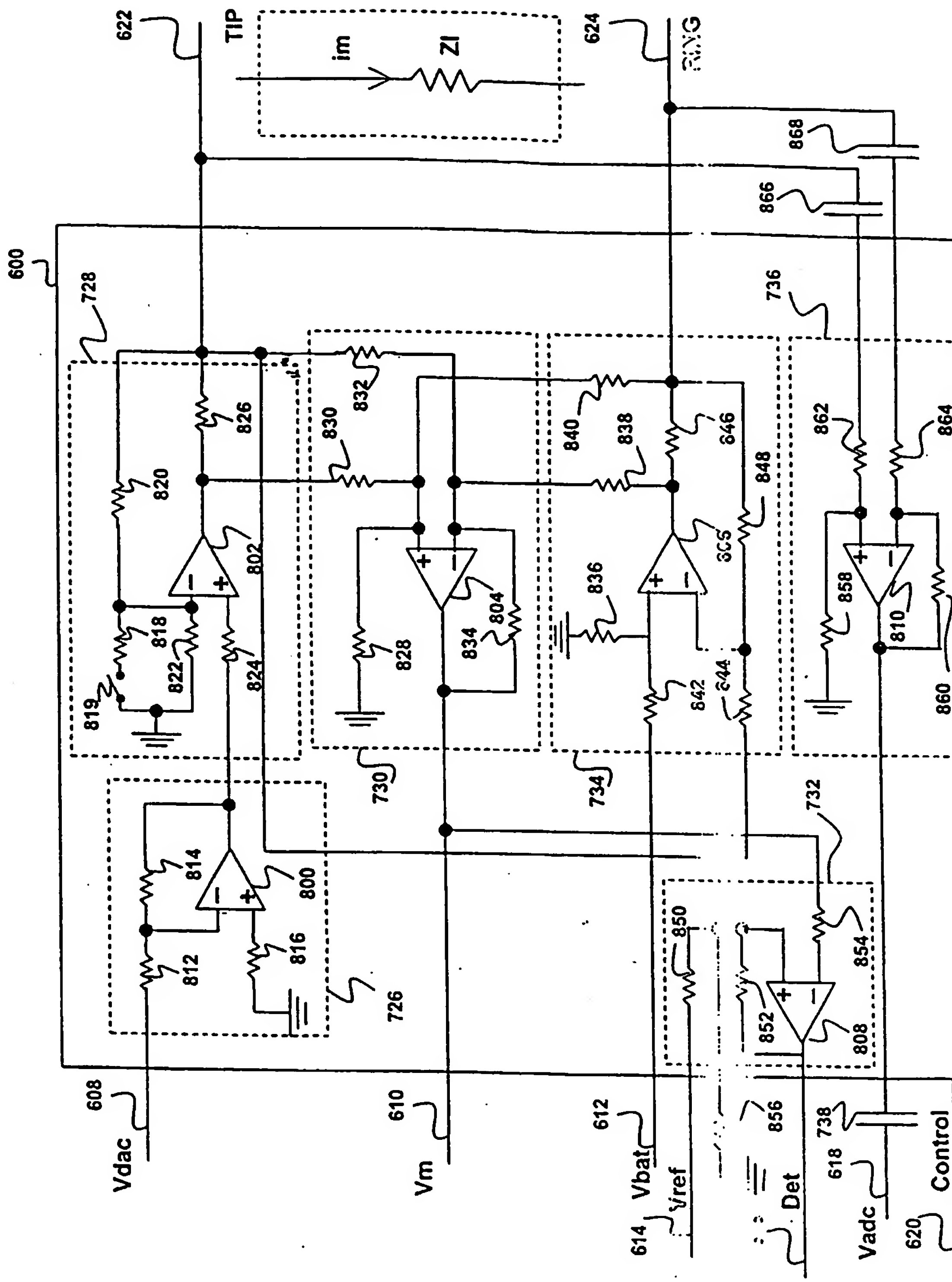

622
TIP
im
Zl
624
RING
868
866
600
728
736
826
832
830
840
838
846
862
864
820
802
848
836
804
858
810
818
822
824
819
828
834
842
844
860
730
734
732
814
800
850
854
812
816
726
852
808
608
610
612
856
738
618
614
620
Vdac
Vm
Vbat
Vref
Det
Vadc
Control

FIG. 48

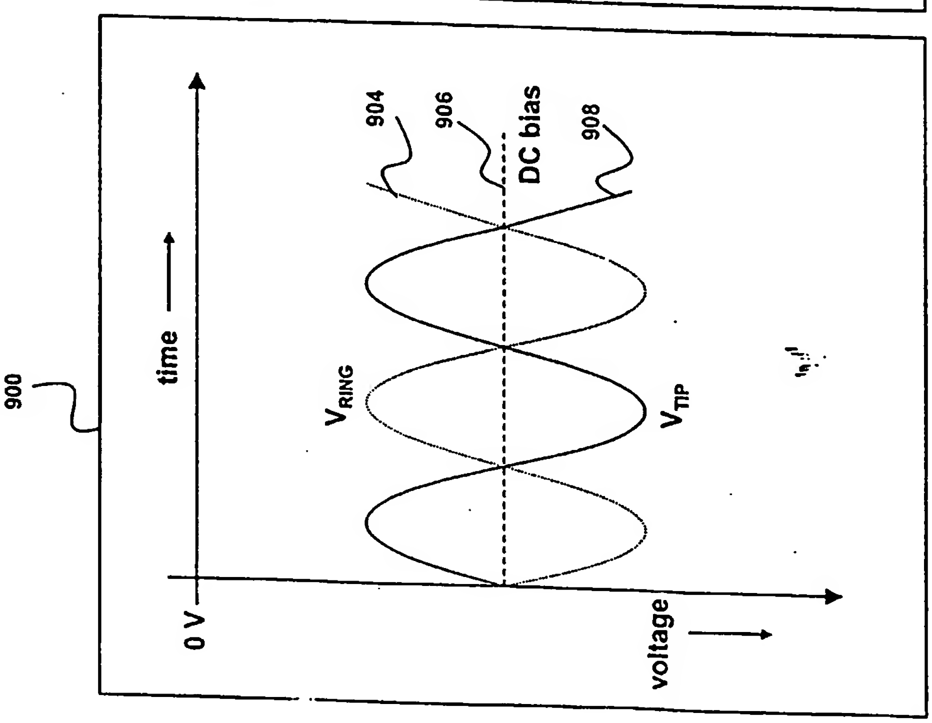


FIG. 49A

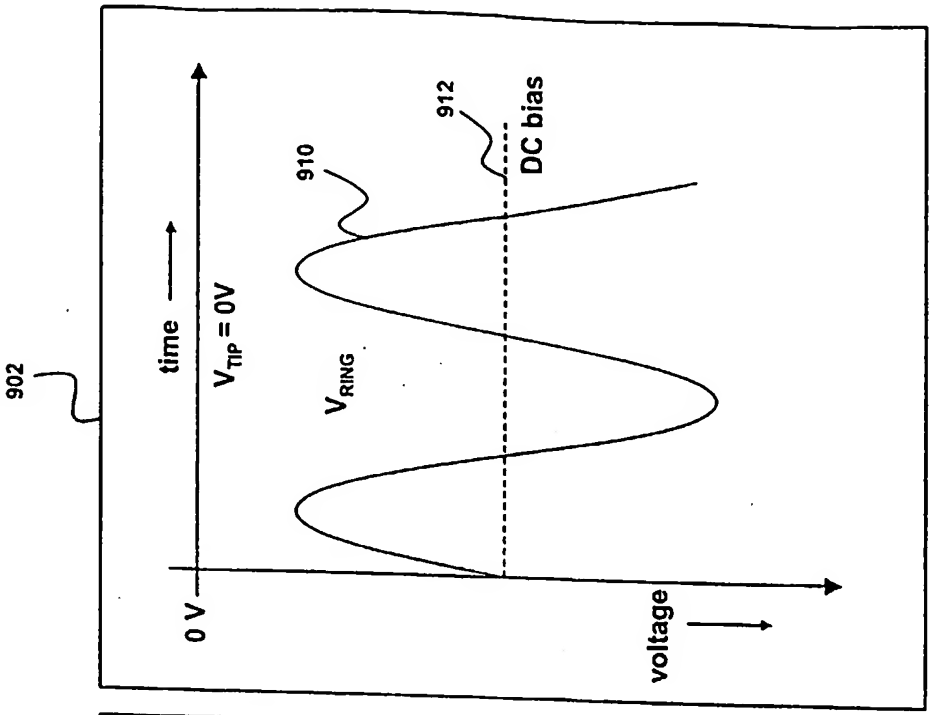


FIG. 49b

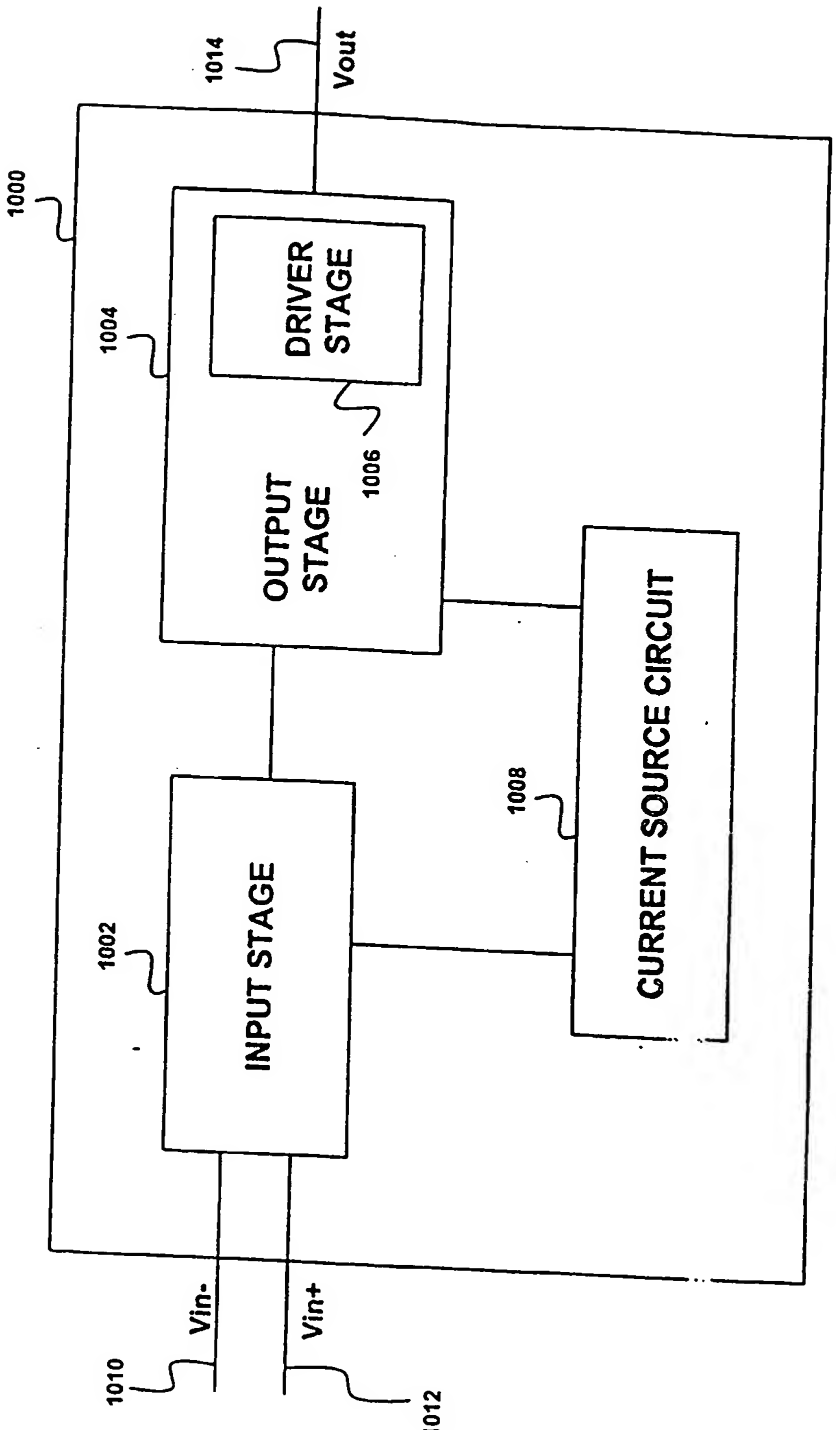

1014
Vout
1000
1004
DRIVER STAGE
1006
OUTPUT STAGE
CURRENT SOURCE CIRCUIT
1008
1002
INPUT STAGE
Vin-
Vin+
1010
1012

FIG. 50

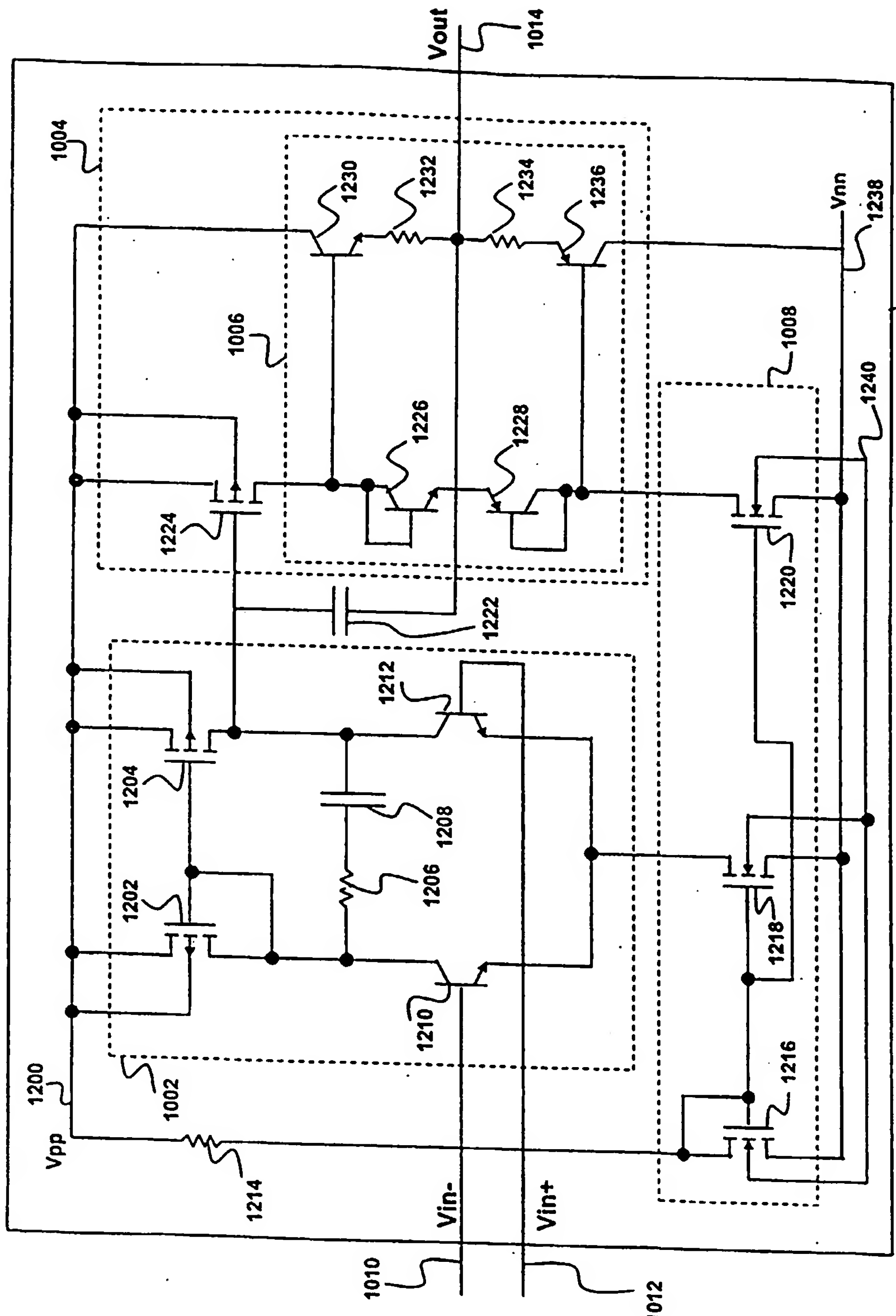

Vout
1014
1004
1006
1230
1232
1234
1236
Vnn
1238
1000
1008
1240
1226
1228
1224
1220
1222
1212
1204
1208
1202
1206
1218
1210
1216
1200
Vpp
1002
1214
Vin-
Vin+
1010
1012

FIG. 51

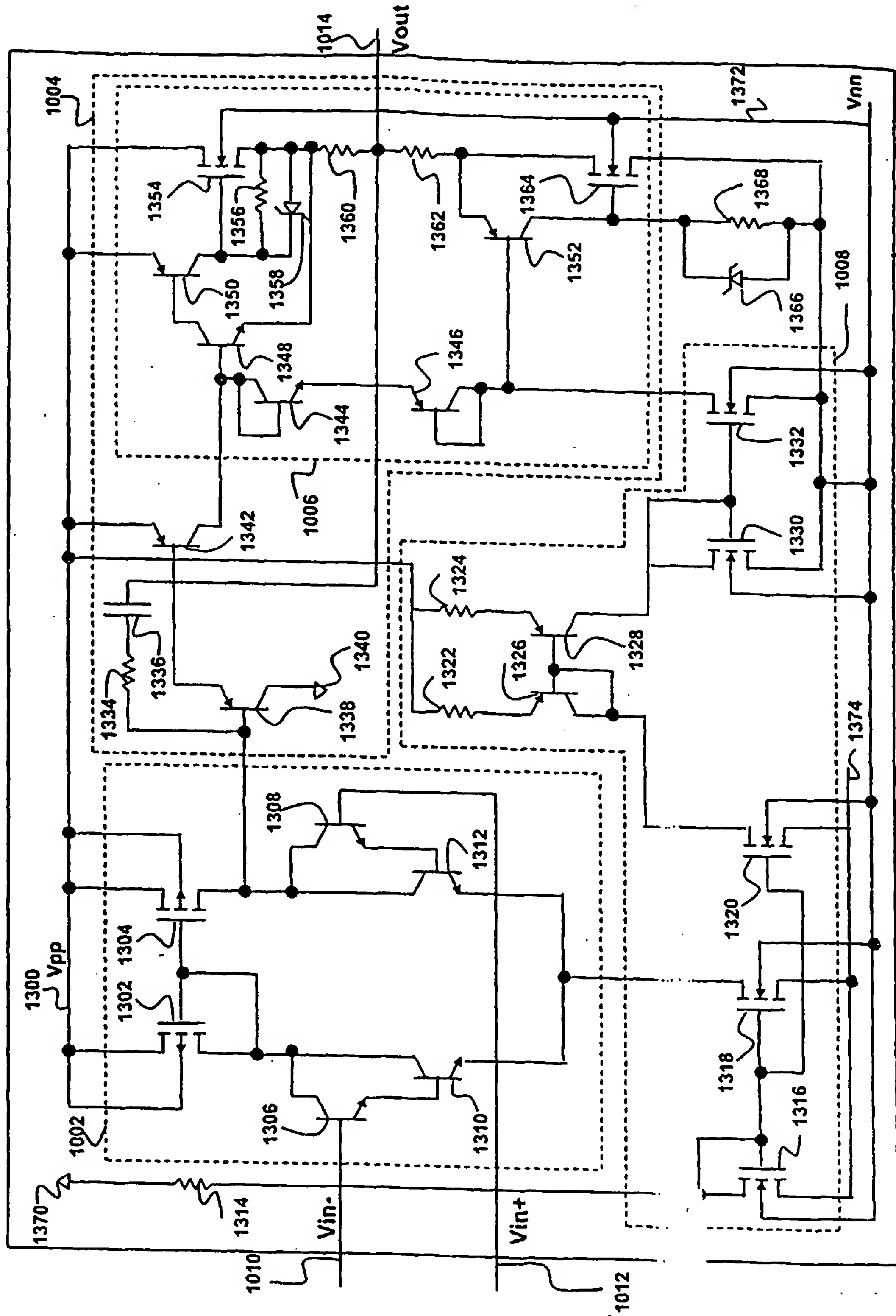

Vout
1014
1004
1006
1008
1000
1002
1300
Vpp
Vnn
Vin-
Vin+
1010
1012
1302
1304
1306
1308
1310
1312
1314
1316
1318
1320
1322
1324
1326
1328
1330
1332
1334
1336
1338
1340
1342
1344
1346
1348
1350
1352
1354
1356
1358
1360
1362
1364
1366
1368
1370
1372
1374

FIG. 52

CLAIMS:

1. A method for synchronizing clocks in a packet transport network, the method comprising:

receiving an external network clock at a central packet network node;

transmitting timing information to a plurality of packet network devices, the timing information based upon the external network clock;

transmitting data that is synchronized to the timing information to the packet network devices;

receiving data that is synchronized to the timing information to the packet network devices; and

delivering packets to an external interface via a packet network that includes data synchronized to the external network clock.

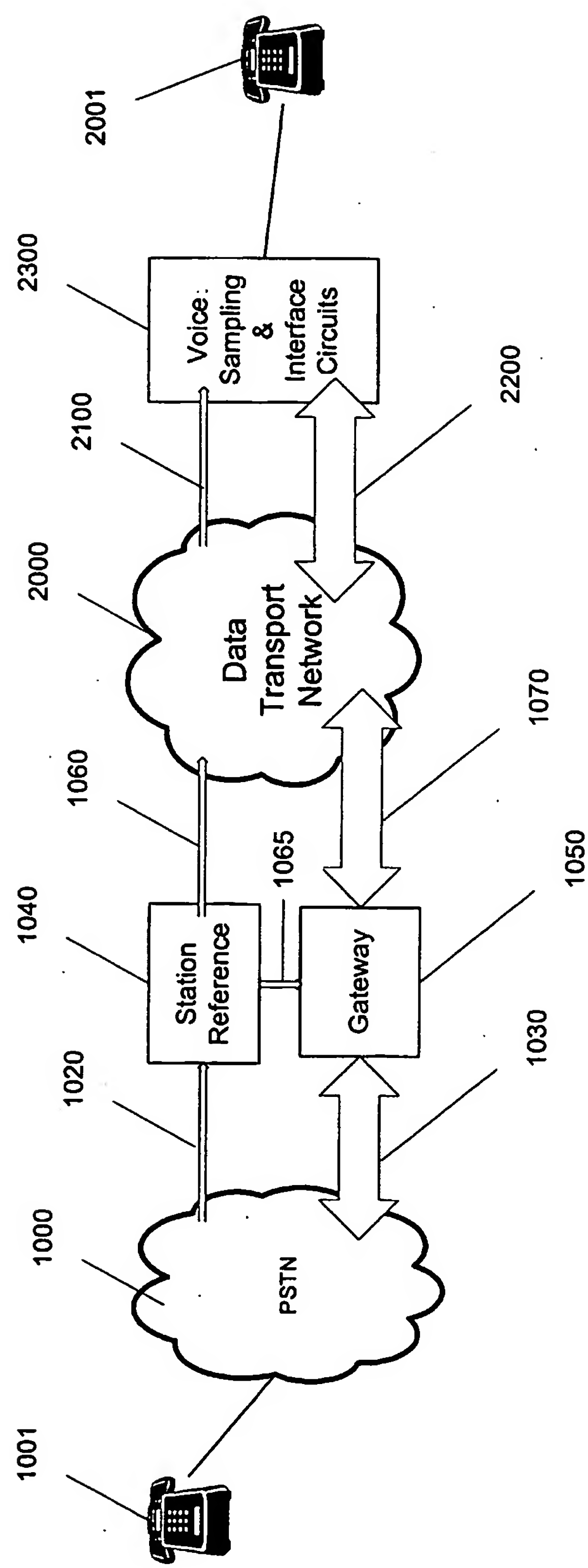

2001
2300
Voice
Sampling
&
Interface
Circuits
2100
2200
2000
Data
Transport
Network
1070
1060
1065
1040
Station
Reference
1050
Gateway
1020
1030
1000
PSTN
1001

Figure 1

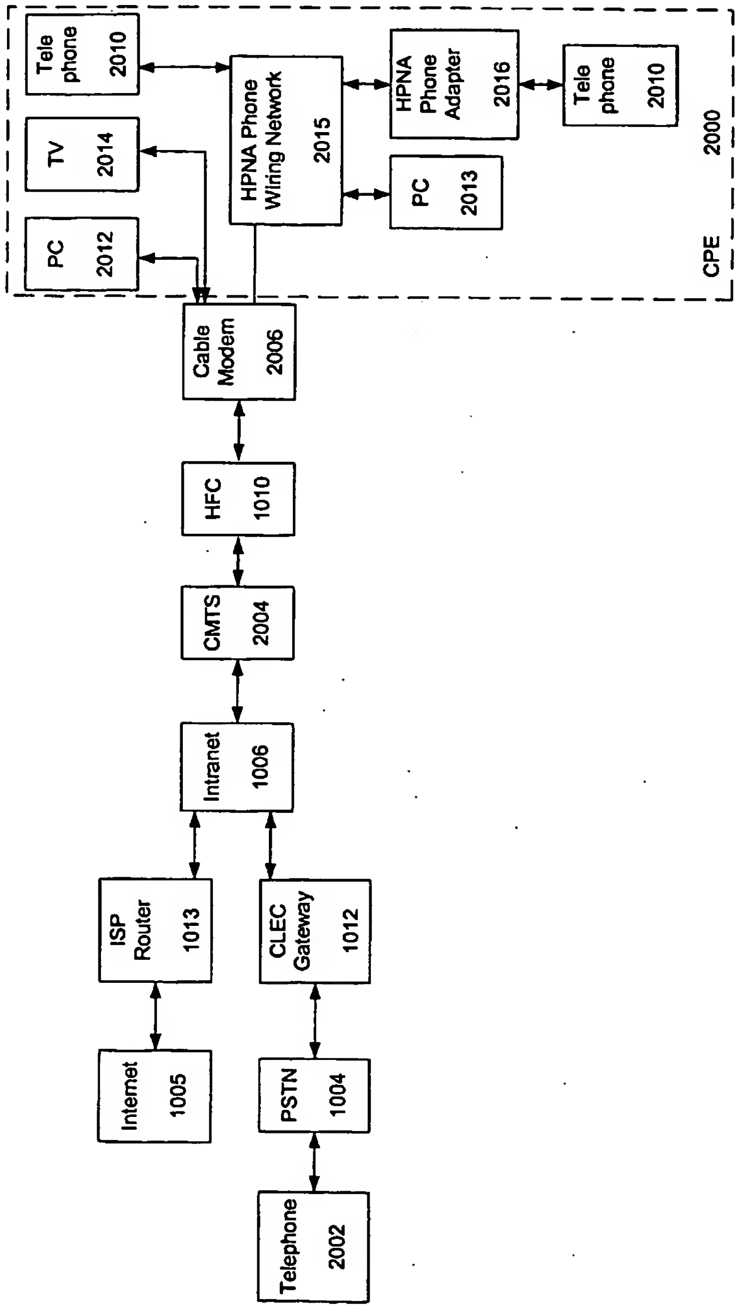

Tele phone 2010
TV 2014
PC 2012
HPNA Phone Wiring Network 2015
HPNA Phone Adapter 2016
Tele phone 2010
PC 2013
CPE 2000
Cable Modem 2006
HFC 1010
CMTS 2004
Intranet 1006
ISP Router 1013
CLEC Gateway 1012
Internet 1005
PSTN 1004
Telephone 2002

FIG. 2

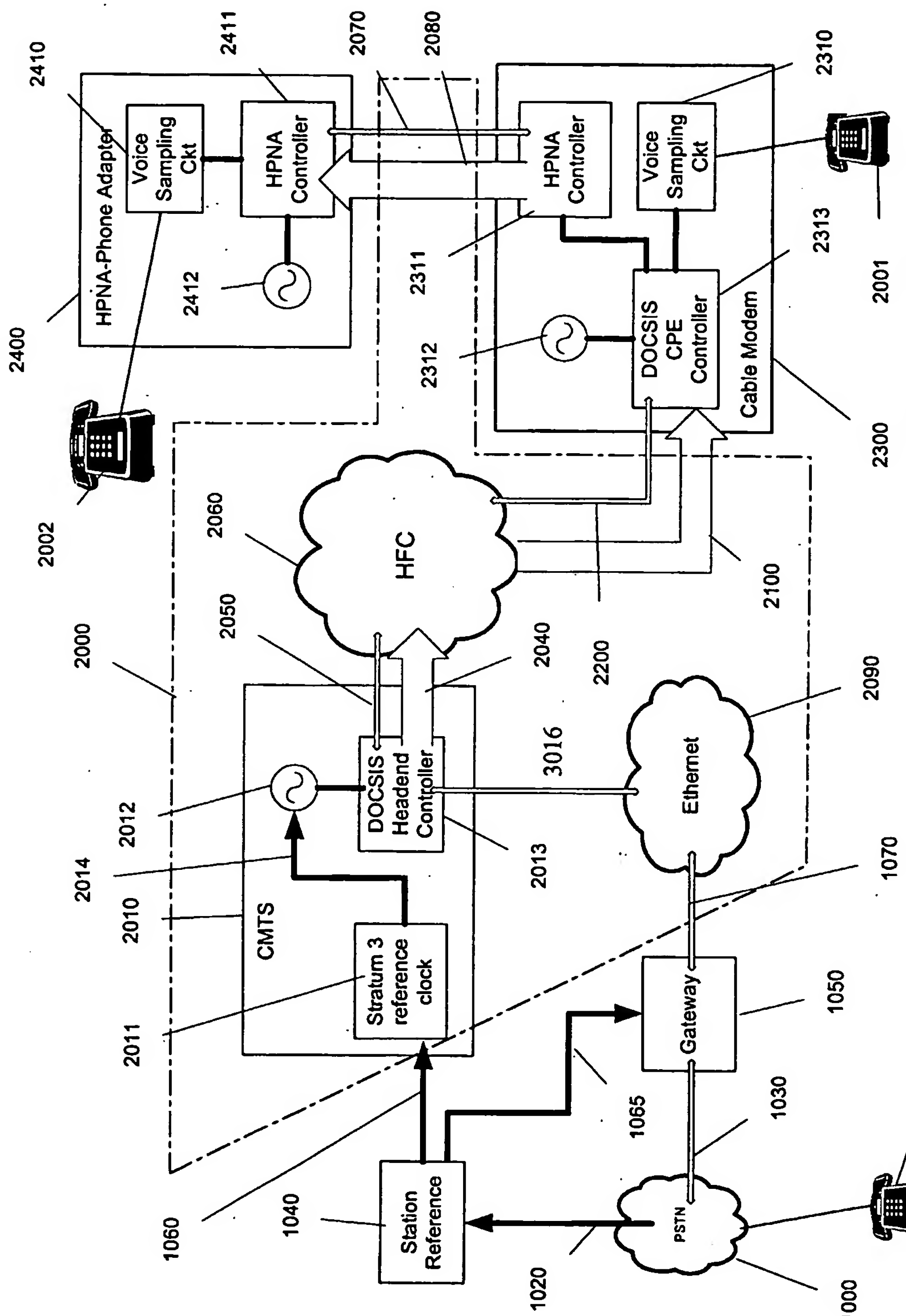

HPNA-Phone Adapter
Voice Sampling Ckt
HPNA Controller
HPNA Controller
Voice Sampling Ckt
DOCSIS CPE Controller
Cable Modem
HFC
CMTS
Stratum 3 reference clock
DOCSIS Headend Controller
Ethernet
Gateway
Station Reference
PSTN
2410
2411
2070
2080
2310
2400
2412
2311
2312
2313
2001
2300
2002
2060
2100
2000
2050
2040
2200
2090
3016
2012
2013
2014
2010
1070
2011
1050
1065
1030
1001
1060
1040
1020
1000

Figure 3

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 00/24405

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04J3/06 H04N7/173 H04L12/28

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04J H04N H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

INSPEC, COMPENDEX, EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 283 079 A (ALCATEL NV ;BELL TELEPHONE MFG (BE)) 21 September 1988 (1988-09-21) column 1, line 48 -column 2, line 3 column 4, line 8 - line 16 column 4, line 31 - line 35	1
Y	US 5 543 951 A (MOEHRMANN KARL-HEINZ) 6 August 1996 (1996-08-06) abstract column 4, line 15 - line 31 column 4, line 46 - line 65 column 6, line 2129 column 7, line 27 - line 49 figures 1-3	1

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
7 February 2001	23/02/2001
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Pieper, T

1

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 98 31115 A (DIVA SYSTEMS CORP) 16 July 1998 (1998-07-16) page 10, line 21 - line 34 page 15, line 29 - line 30 page 20, line 8 - line 14 figure 1 ---	1
A	SDRALIA V ET AL: "PERFORMANCE CHARACTERISATION OF THE MCNS DOCSIS 1.0 CATV PROTOCOL WITH PRIOTITISED FIRST COME FIRST SERVED SCHEDULING" IEEE TRANSACTIONS ON BROADCASTING,US,IEEE INC. NEW YORK, vol. 45, no. 2, June 1999 (1999-06), pages 196-205, XP000851909 ISSN: 0018-9316 page 198, left-hand column, last paragraph -right-hand column, paragraph 5 page 199, right-hand column, line 6 - line 40 page 201, left-hand column, paragraph 1 ---	1
P,X	NN: "Cisco Cable Clock Card for the Cisco uBR7246 VXR Universal Broadband Router" INTERNET PUBLICATION, 'Online! pages 1-22, XP002159663 26-03-2000 Retrieved from the Internet: <URL:http://cisco.com/univercd/cc/td/doc/product/cable/cab_r_sw/natlclck.pdf> 'retrieved on 2001-01-29! page 1 -page 3 -----	1

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

In tional Application No
PCT/US 00/24405

Patent document cited in search report		Publication date	Patent family member(s)			Publication date
EP 0283079	A	21-09-1988	BE	1000414	A	22-11-1988
			AT	107446	T	15-07-1994
			AU	602958	B	01-11-1990
			AU	1262988	A	22-09-1988
			CA	1282149	A	26-03-1991
			CN	88101402	A,B	05-10-1988
			DE	3850156	D	21-07-1994
			DE	3850156	T	24-11-1994
			ES	2056896	T	16-10-1994
			FI	881187	A,B,	19-09-1988
			JP	63290431	A	28-11-1988
			KR	9609461	B	19-07-1996
			MX	167498	B	25-03-1993
			US	4881223	A	14-11-1989
US 5543951	A	06-08-1996	EP	0673166	A	20-09-1995
			AT	180937	T	15-06-1999
			DE	59408360	D	08-07-1999
WO 9831115	A	16-07-1998	AU	5693798	A	03-08-1998
			EP	0951681	A	27-10-1999
			US	6166730	A	26-12-2000